79-855

No.

Supreme Court, U. S.
FILED
DEC 3 1979
MICHAEL RODAK, JR., CLERK

In the Supreme Court of the United States

OCTOBER TERM, 1979

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS, PETITIONER

v.

JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN

PETITION FOR A WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

WADE H. MCCREE, JR.
Solicitor General

JOHN H. SHENEFIELD
Assistant Attorney General

ROBERT B. NICHOLSON
FREDERIC FREILICHER
Attorneys
Department of Justice
Washington, D.C. 20530

JOSEPH F. NAKAMURA
Solicitor

THOMAS E. LYNCH
Associate Solicitor
United States Patent and Trademark Office
Washington, D.C. 20231

INDEX

	Page
Opinions below	1
Jurisdiction	2
Question presented	2
Statute involved	2
Statement	2
Reasons for granting the petition	8
Conclusion	14
Appendix A	1a
Appendix B	16a
Appendix C	17a
Appendix D	31a
Appendix E	42a
Appendix F	52a

CITATIONS

Cases:

Bergy, In re, 596 F.2d 952, cert. granted, No. 79-136 (Oct. 29, 1979)	9
Deepsouth Packing Co. v. *Laitram Corp.*, 406 U.S. 518	13
Diehr, In re, 602 F.2d 982, reh. denied (Oct. 18, 1979)	13
Eibel Process Co. v. *Minnesota & Ontario Paper Co.*, 261 U.S. 45	11
Flook, In re, 559 F.2d 21	8
Gelnovatch, In re, 595 F.2d 32	12
Ghiron, In re, 442 F.2d 985	3

Cases—Continued Page

Gottschalk v. *Benson*, 409 U.S. 63.....6, 7, 8, 10, 11, 12
Honeywell, Inc. v. *Lithonia Lighting, Inc.*, 317 F.Supp. 406 3
Johnson, In re, 589 F.2d 1070 12
Maucorps, In re, No. 79-554 (C.C.P.A. Nov. 1, 1979), 453 Pat. T.M. & Copyright J. (BNA) A-1 (Nov. 8, 1979).... 11
O'Reilly v. *Morse*, 56 U.S. (15 How.) 61 12
Parker v. *Flook*, 437 U.S. 584.....6,7, 8, 9, 10, 11 12, 13, 14
Sarkar, In re, 588 F.2d 1330 12
Toma, In re, 575 F.2d 872 12
Waldbaum, In re, 457 F.2d 997 3

Statutes:

28 U.S.C. 1338 13
35 U.S.C. 101 .....2, 5, 8, 10
35 U.S.C. 281 *et seq.* 13

Miscellaneous:

Cunningham & Smith, "Computer Services: A Menu of Options," *Datamation* (May 25, 1979) 3
P. Jordain & M. Breslau, *Condensed Computer Encyclopedia* (1969) 4
Ross, *The Patentability of Computer "Firmware,"* 59 J. Pat. Off. Soc'y 731 (1977) 3, 4
C. Sippl & C. Sippl, *Computer Dictionary and Handbook* (2d ed. 1972) 3, 6

In the Supreme Court of the United States

OCTOBER TERM, 1979

No.

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS, PETITIONER

v.

JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN

PETITION FOR A WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

The Solicitor General, on behalf of the Commissioner of Patents and Trademarks, petitions for a writ of certiorari to review the judgment of the United States Court of Customs and Patent Appeals in this case.

OPINIONS BELOW

The opinion of the Court of Customs and Patent Appeals (App. A, *infra*, 1a-15a) is reported at 600 F.2d 807. The opinion of the Patent and Trademark Office Board of Appeals (App. C, *infra*, 17a-30a) and the opinions of the patent examiner (App. D, *infra*, 31a-41a) are not reported.

JURISDICTION

The judgment of the Court of Customs and Patent Appeals was entered on July 5, 1979 (App. B, *infra*, 16a). On October 2, 1979, the Chief Justice extended the time to file a petition for a writ of certiorari to and including December 2, 1979. The jurisdiction of this Court is invoked under 28 U.S.C. 1256. *Gottschalk* v. *Benson*, 409 U.S. 63 (1972); *Dann* v. *Johnson*, 425 U.S. 219 (1976).

QUESTION PRESENTED

Whether a computer program that regulates the internal operation of a computer is patentable subject matter under 35 U.S.C. 101.

STATUTE INVOLVED

35 U.S.C. 101 provides:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

STATEMENT

On April 21, 1975, John J. Bradley and Benjamin S. Franklin applied for a patent (App. E, *infra*, 42a-51a) on an invention that directs the transfer of data within a computer.[1]

---

[1] The invention is assigned to Honeywell Information Systems, Inc. Brief for Appellants at 2.

The computer's hardware [2] includes a main memory which has a system base [3] with which the computer programmer can communicate.[4] In high performance computers, where speed of operation is important, certain information that is part of the system base may be stored in storage components of the computer called "scratchpad registers" (App. A, *infra*, 2a & n.1). Because these registers are not, however, normally accessible to the programmer by means of com-

---

[2] The physical components of the computer are called "hardware." *Honeywell, Inc.* v. *Lithonia Lighting, Inc.*, 317 F.Supp. 406, 408 (N.D. Ga. 1970); Ross, *The Patentability of Computer "Firmware,"* 59 J. Pat. Off. Soc'y 731, 736 (1977).

[3] The system base controls the operation of the computer; it contains the process information used by the computer to carry out programmed operations.

[4] A programmer wanting a general purpose computer to perform a certain assigned task communicates with the computer by translating the task into coded instructions that can be read, understood and executed by the computer system. These instructions, called a "program," are written by the programmer in special programming language, such as COBOL or FORTRAN. See C. Sippl & C. Sippl, *Computer Dictionary and Handbook* 80, 191 (2d ed. 1972); *In re Ghiron*, 442 F.2d 985, 986 (C.C.P.A. 1971); *In re Waldbaum*, 457 F.2d 997, 998 (C.C.P.A. 1972). Computer programs as well as the information (data) fed into the computer are commonly called "software." *Honeywell, Inc.* v. *Lithonia Lighting, Inc., supra*, 317 F.Supp. at 408; Ross, *supra*, 59 J. Pat. Off. Soc'y at 736. Computer users either write their own programs or obtain them from software vendors. In 1978, user expenditures for software products amounted to $981 million. Cunningham & Smith, "Computer Services: A Menu of Options," Datamation 89, 91 (May 25, 1979).

puter programs (App. A, *infra*, 3a), changing the data they contain has been a problem.[5]

Bradley's claimed invention provides an improved method for changing the data in the registers. In addition to certain hardware elements, it employs "firmware" (*ibid.*). Firmware consists of a sequence of basic instructions stored as a microprogram in the computer's control memory.[6] When activated by software, the instructions cause a particular sequence

---

[5] To change the data in the scratchpad registers the programmer has had to reprogram the entire system base, a time consuming task, or use software limited to that particular computer model, an option unacceptable to many computer users (App. A, *infra*, 3a; App. C, *infra*, 18a-19a).

[6] A microprogram has been defined as "a sequence of elementary steps which permits the computer hardware to carry out a program instruction." Ross, *supra*, 59 J. Pat. Off. Soc'y at 754, quoting from P. Jordain & M. Breslau, *Condensed Computer Encyclopedia* 318 (1969). Thus, the microprogrammer, like the programmer, uses special programming language to prepare instruction for the computer to carry out in sequence. Ross, *supra*, 59 J. Pat. Off. Soc'y at 765. The microprogram is usually stored in a "control memory" physically separate from other computer memories. *Id.* at 758. Early control memories were of the "read only" variety (ROM) which meant that the microprogram was physically fixed in the control memory by the manufacturer. To change the microprogram the manufacturer had to install a new ROM with the new microprogram. Technical advances made it possible for the user himself to microprogram the control memory (programmable read only memory, PROM). Although the original PROMs could be programmed only once, many large computers now have a "read-write" PROM, which permits the user to reprogram the control memory any number of times electrically. *Id.* at 760-761.

of computer operations to take place.[7] An instruction in the firmware developed by Bradley permits the programmer to communicate with the scratchpad registers and to switch data back and forth from the registers to the system base in the main memory (App. C, *infra*, 19a).[8]

The patent examiner rejected the invention as drawn to nonstatutory subject matter under 35 U.S.C. 101 (App. D, *infra*, 34a, 37a). He noted that the hardware arrangement to which the claims referred —"a main memory, central processing unit, and scratchpad registers"—were "wellknown components" that were "admittedly old in the art * * *" (*id.* at 38a).[9] He found that the only novel aspect

---

[7] Bradley's firmware is stored in a control memory of the PROM type, which can be microprogrammed by the user (App. E, *infra*, 45a-46a). The record does not clearly indicate which type of PROM is involved, although Bradley does state that the PROM can be "reloaded" (*id.* at 45a).

[8] When the programmer activates what Bradley calls a "switch system base instruction," this initiates the execution of a microprogram that transfers the data (App. C, *infra*, 19a). In Bradley's specification, the switch system base instruction is disclosed by the diagram in Figure 15a (App. E, *infra*, 48a) and the microprogram steps are disclosed in the flow charts of Figures 15b and 15c (*ibid.*).

[9] The examiner further noted in his answer before the Patent and Trademark Office Board of Appeals that microprogrammed control units like Bradley's were also old in the art (App. C, *infra*, 23a). Thus, he concluded that the invention "resides in an instruction in firmware which changes upon execution, information in the system base" (*ibid.*).

of the invention resided in an algorithm [10] designed to control the multiprogramming computer to solve the particular problem indicated (App. D, *infra*, 38a-39a). He concluded that this "program implemented algorithm" was not patentable under *Gottschalk* v. *Benson*, 409 U.S. 63 (1972) (*id.* at 39a).

The Patent and Trademark Office Board of Appeals ("Board") agreed with the examiner (App. C, *infra*, 17a-30a). It found that "save for the microprogramming" Bradley's arrangement of hardware was old in the art (*id.* at 26a-27a). It ruled that the fact that Bradley's claims were in "apparatus," rather than "methods," format did not make them any less "related to an algorithm" than was the claim in *Benson* (*id.* at 28a-29a).[11] Added the Board, "[a] claim for an improved method of calculation, even when tied to a specific end use, is unpatentable" under *Parker* v. *Flook*, 437 U.S. 584 (1978) (*id.* at 29a-30a). The Board concluded that Bradley's claimed invention con-

---

[10] In computer terminology, an algorithm has been defined as follows:

> 1. A fixed step-by-step procedure for accomplishing a given result; usually a simplified procedure for solving a complex problem, also a full statement of a finite number of steps. 2. A defined process or set of rules that leads and assures development of a desired output from a given input. A sequence of formulas and/or algebraic/logical steps to calculate or determine a given task; processing rules.

C. Sippl & C. Sippl, *supra*, at 23.

[11] Thus, the claims "recite[d] apparatus as a matter of form only" (*id.* at 30a).

sists of apparatus that is old in the art coupled with "programming, which is nonstatutory" under *Benson* and *Flook* (App. C, *infra*, 30a).

The Court of Customs and Patent Appeals ("CCPA") reversed. It observed that Bradley was claiming "a machine or apparatus" composed of a "combination of hardware elements" which fell "literally within the boundaries of § 101" (App. A, *infra*, 12a). The court recognized that this "combination" also involves "a portion of the computer's control store microprogrammed in a particular manner." It left standing the Board's determination that the apparatus was old in the art and that only the microprogram was new (*id.* at 13a), but it rejected the Board's "distillation of [the] claim down to the information contained in the firmware * * *" (*id.* at 14a). The proper analysis, said the court, involves determining "whether the claim recites a mathematical algorithm, and, if so, whether it preempts the use of the algorithm" (*ibid.*). Applying this test, it stated that there was no mathematical algorithm here (*ibid.*). Although certain calculations were made, they did not "transform the invention as a whole into a method of calculation" because, after the invention's task was completed, there was not present a "solution of an equation, such as the new alarm limit in *Flook*, or the equivalent pure binary number, as in *Benson* * * *" (*id.* at 14a-15a).

REASONS FOR GRANTING THE PETITION

In concluding that Bradley's invention is patentable subject matter under 35 U.S.C. 101, the Court of Customs and Patent Appeals refused to apply the analysis of the claims required under *Parker* v. *Flook*, 437 U.S. 584 (1978), read too narrowly this Court's decision in *Gottschalk* v. *Benson*, 409 U.S. 63 (1972), and thus significantly extended coverage of the patent laws without legislative authorization.

1. This Court's decision in *Parker* v. *Flook*, 437 U.S. 584 (1978), requires that a patent claim involving an algorithm be analyzed under Section 101 in two steps. First, one must determine whether the claim contains a principle, formula, idea or concept which, as one of the "basic tools of scientific and technological work," is itself unpatentable and must be separated from the rest of the claim. *Id.* at 591-594. Second, what remains of the claim must then be analyzed to determine whether it is old in the art. *Id.* at 594-595.

The Board of Appeals followed this method of analysis, found that everything in Bradley's claim was old in the art except the programmed algorithm, and held the algorithm itself was not patentable. The court below, however, adhered to the approach it had developed in pre-*Flook* cases (App. A, *infra*, 10a; see also *In re Flook*, 559 F.2d 21 (C.C.P.A. 1977)). It did not dispute the Board's findings that everything in Bradley's claim was old except the program [12]

[12] Bradley had claimed that this finding was erroneous. This is surprising, since Bradley's arrangement of computer hardware incorporated by reference an earlier application

and that the program expressed an algorithm (App. A, *infra*, 13a-14a). It nonetheless rejected the Board's analysis as involving a "distillation of appellants' claim down to the information contained in the firmware * * *" (*id.* at 14a). It insisted that, since Bradley had drafted the claim as one for an an arrangement of hardware, it must be so regarded "as a whole" and involved neither a "mathematical algorithm" nor "calculations" which transform the invention "into a method of calculation" (*ibid.*).

The CCPA's analysis is squarely at odds with *Flook*.[13] What the court labelled an inappropriate "distillation" of Bradley's claim is precisely the kind of two-step analysis that this Court mandated in *Flook*. The lower court's insistence upon discarding that analysis for reliance on the inventor's characterization of his claim as an arrangement of hardware makes patentability turn on the skill of the draftsman and would "ill serve the principles underlying the prohibition against patents for 'ideas' or

---

for a system base admittedly invented by Claude Carre and others (App. E, *infra*, 42a, 47a). The Carre drawings disclose an arrangement of hardware identical to Bradley's. Thus, 21 of Bradley's 22 sheets of drawings are identical to Carre's. Compare Bradley's summary of the drawings, Figures 1-14i (App. E, *infra*, 43a-44a) with Carre's summary of Figures 1-14i (App. F, *infra*, 52a-53a). Only Bradley's algorithm is new. The court below avoided this issue (App. A, *infra*, 12a).

[13] This is not the first time that that court has rejected *Flook*. See *In re Bergy*, 596 F.2d 952, 962-964, 966 (C.C.P.A. 1979), cert. granted, No. 79-136 (Oct. 29, 1979). See also pages 12-13, *infra*.

phenomena of nature." *Flook*, *supra*, 437 U.S. at 593. Bradley's claim, however artfully described, seeks a patent on an algorithm,[14] for everything else in his claim is old and not his inventive contribution.[15] Finally, the court's reference to the claim "as a whole" fails to focus on the algorithm itself, as required by *Flook*, and reflects instead the approach of the dissent in *Flook*, *supra*, 437 U.S. at 599-600.

The court compounded these errors by assuming that, so long as the algorithm was not mathematical, its patentability under Section 101 posed no problems (App. A, *infra*, 7a-11a, 14a). Undoubtedly, a claim whose only novel element is a computer program expressing a mathematical algorithm is not patentable subject matter. *Benson* and *Flook* make that clear. But although the algorithms in *Flook* and *Benson* were mathematical, the Court's holdings did not rest on any distinction between mathematical and non-mathematical algorithms. The Court broadly stated in *Benson* that "[p]henomena of nature, though just discovered, mental processes, and abstract intellectual

---

[14] Bradley's abstract of disclosure makes this plain:

> A model-independent instruction in firmware is provided to change, upon execution, information in the system base which is located in both main memory and internal scratch pad registers.

App. E, *infra*, 42a.

[15] Thus the lower court's statement that "[i]f appellants were claiming the information embodied in the firmware or the firmware itself, per se, a different case would be presented [and that] [w]e express no opinion on the statutory nature of such an invention * * *" (App. A, *infra*, 13a) blinks reality.

concepts are not patentable, as they are the basic tools of scientific and technological work." 409 U.S. at 67. The Court in *Flook* had the same broad focus when it stated that "[d]ifficult questions of policy concerning the kinds of programs that may be appropriate for patent protection and the form and duration of such protection can be answered by Congress on the basis of current empirical data not equally available to this tribunal." 437 U.S. at 595 (footnote omitted).

Phenomena of nature, mental processes and abstract intellectual concepts may be mathematical, but they need not be. Indeed, most ideas and concepts are not mathematical, yet the absence of mathematical expression does not make them any more the subject of the patent laws. The phenomena that water runs downhill and that the sun rises in the east are no more patentable than mathematical equations. Cf. *Eibel Process Co.* v. *Minnesota & Ontario Paper Co.*, 261 U.S. at 45 (1923), cited in *Parker* v. *Flook*, *supra*, 437 U.S. at 590 n.12.

So it is with computer programs.[16] The proper inquiry is whether the program, be it mathematical

---

[16] The fact that Bradley's program is firmware rather than classic software is immaterial. As the CCPA noted, firmware "may be likened to software * * *" (App. A, *infra*, 13a). This particular firmware can be programmed by the user (App. E, *infra*, 45a-46a; note 7, *supra*). But even if the program were permanently built into the machine, this would not be a matter of particular significance, as the CCPA has elsewhere recognized. See *In re Maucorps*, No. 79-554 (C.C.P.A. Nov. 1, 1979), 453 Pat. T.M. & Copyright J. (BNA) A-1 (Nov. 8, 1979).

or nonmathematical, expresses a phenomenon of nature, mental process, or an abstract intellectual concept. Bradley's program, when measured against this standard, is no more patentable than the algorithms involved in *Benson* and *Flook*. Like programs generally, Bradley's program is a set of directions to the computer. It commands the switching of data, of whatever type, untied to any particular end use. Although the directions are for the movement of information in the computer system base, they as much reflect abstract intellectual concepts as directions for the translation of texts from Russian to English (*In re Toma*, 575 F.2d 872 (C.C.P.A. 1978)) or for a hitherto unknown and faster route between Washington and New York. The computer programmer simply uses the computer to implement the idea embodied in the program. See *Benson*, *supra*, 409 U.S. at 67; *O'Reilly* v. *Morse*, 56 U.S. (15 How.) 62, 112-113 (1853).

2. The importance of this case transcends its particular facts and the lower court's evident unwillingness to apply *Parker* v. *Flook*. That unwillingness is confirmed by a number of other cases in which the CCPA has similarly adhered to the analysis of the dissent in *Flook*, rather than the opinion of the Court. *In re Gelnovatch*, 595 F.2d 32, 36-37 (1979); *In re Johnson*, 589 F.2d 1070, 1081 (1978); *In re Sarkar*, 588 F.2d 1330, 1332-1333 (1979). The CCPA's interpretation of *Benson* and *Flook* is highlighted in

In re Diehr, 602 F.2d 982 (C.C.P.A. 1979), reh. denied (Oct. 18, 1979), a case involving a mathematical algorithm similar to the one involved in *Flook*. We plan to file a petition for a writ of certiorari in *Diehr* shortly. We suggest that it be considered with this case, and that it would be desirable for the two cases either to be consolidated for argument in this Court or to be heard in tandem.[17]

In light of the consistent rejection of *Flook* by the court below, review by this Court is of prime importance to efficient administration of the patent system. The Patent and Trademark Office presently has pending more than 3000 patent applications in which the patentability of computer software or firmware is a potential issue. Some 1200 applications involve mathematical algorithms; over 1800 involve non-mathematical algorithms. Unless this Court grants review now, the CCPA's approach, exemplified by the decision below, puts the Commissioner in an untenable position. Should he follow the analysis required by the CCPA, he would be untrue to *Flook*, would usurp the congressional function by expanding the scope of the patent laws (see *Deepsouth Packing Co.* v. *Laitram Corp.*, 406 U.S. 518, 531 (1972)) and withal would settle nothing in the law, for the issues of patentability would remain to be litigated in infringement suits (see 35 U.S.C. 281 *et seq.*; 28 U.S.C. 1338). On the other hand, should he continue to adhere to his view that computer programs are themselves un-

---

[17] We will send respondents a copy of our petition in *Diehr* as soon as it is filed.

patentable, there is the gloomy prospect of years of repeated, costly and unsuccessful litigation in the Court of Customs and Patent Appeals. This Court in *Flook* strongly intimated that computer programs are not patentable under current law. See 437 U.S. at 595. A definitive ruling to that effect would correctly leave to Congress the decision whether and how to extend the patent laws to computer programs, would greatly facilitate administration of the patent system, and would forestall the unsettling effects of a proliferation in the economy of unauthorized (and ultimately invalid) patents on computer programming.

CONCLUSION

For the foregoing reasons, the petition for a writ of certiorari should be granted.

Respectfully submitted.

WADE H. MCCREE, JR.
Solicitor General

JOHN H. SHENEFIELD
Assistant Attorney General

ROBERT B. NICHOLSON
FREDERIC FREILICHER
Attorneys

JOSEPH F. NAKAMURA
Solicitor

THOMAS E. LYNCH
Associate Solicitor
United States Patent
and Trademark Office

DECEMBER 1979

APPENDIX A

UNITED STATES COURT OF
CUSTOMS AND PATENT APPEALS

Appeal No. 79-533.

IN THE MATTER OF THE APPLICATION
OF
JOHN J. BRADLEY and BENJAMIN S. FRANKLIN

Serial No. 570,331.

Decided: July 5, 1979

Before MARKEY, Chief Judge, RICH, BALDWIN, and MILLER, Associate Judges, and COWEN,* Senior Judge.

RICH, Judge.

This appeal is from the decision of the Patent and Trademark Office (PTO) Board of Appeals (board) affirming the rejection of claims 1-6, all of the claims in appellants' application serial No. 570,331, filed April 21, 1975, for "Switch System Base Mechanism," as being drawn to subject matter which is non-statutory under 35 USC 101. We reverse.

The Invention

Appellants' invention is in the field of computer technology. It does not relate to computer applica-

---

* The Honorable Wilson Cowen, U.S. Court of Claims, sitting by designation.

tions, i.e., any specific task that a computer is asked to perform, but rather to the internal operation of the computer and its ability to manage efficiently its operation in a multiprogrammed format. A multiprogrammed format is one in which the computer is capable of executing more than one program, and thus perform more than one application at the same time, without the need to reprogram the computer for each task it must perform.

Specifically, the invention relates to altering or repositioning information in the computer's system base. The system base of a computer is a fixed area in main memory which acts as the root for all information structures in the computer. In high performance computer systems, it is very advantageous to store portions of the system base in "scratchpad" registers [1] located in the central processing unit (CPU) of the computer rather than in main memory. This greatly enhances the speed of operation of the computer because the access time (the time it takes to retrieve information from a given place) associated with the scratchpad registers is less than that associated with main memory. A problem arises, however, because a programmer may wish to change the positions or content of information in the system base which happens to be located in the scratchpad

---

[1] A scratchpad register, also known as a scratchpad memory, is a plurality of multibit storage locations, usually located in the central processing unit (CPU) of a computer, used for temporary storage of program information, operands, and calculation results for use by the computer's arithmetic and logic unit, and other information of a temporary nature.

registers. These registers are "invisible" to the programmer since, unlike main memory, they cannot be accessed by software (computer programs). Prior art systems altered the system base information resident in the scratchpad registers by either reinitializing the system base (completely reloading the system base containing the new information), a process which consumes a considerable amount of time, or by using software which takes advantage of the model-dependent properties of the particular computer. The latter method has the undesirable effect of resorting to reliance on model-dependent software, which is unacceptable to some computer users.

Appellants' invention enables system base information to be altered without having to resort to these techniques and their accompanying drawbacks. They accomplish their result by employing a "firmware" module, consisting of hardware elements permanently programmed with a microcode, which directs the data transfers, between the scratchpad registers and the system base located in main memory, which are necessary to effect the alteration.

Claim 1 is representative of the appealed claims:

> 1. In a multiprogramming computer system having a main memory, a central processing unit (CPU) coupled to said main memory, said (CPU) controlling the state of a plurality of groups of processes being in a running, ready, wait or suspended state, said computer system also having scratchpad registers being accessible to an operating system for controlling said multiprogramming computer system, a data structure

for storing coded signals for communicating between said processes and said operating system, and said scratchpad registers, said data structure comprising:

(a) first means in said data structure and communicating with said operating system for storing coded signals indicative of an address for a selected one of said processes;

(b) second means in said first means for storing coded signals indicating priority of said selected one of said processes in relation to others of said processes for obtaining control of said CPU when ready;

(c) third means in said data structure and communicating with said operating system, for storing coded signals indicative of an address for a selected one of said plurality of groups of processes, and,

(d) fourth means coupled to said data structure and said scratchpad registers, for generating signals causing the changing of information in said data structure and said scratchpad registers.

The Rejection

The examiner rejected the appealed claims on the authority of *Gottschalk* v. *Benson*, 409 U.S. 63, 175 USPQ 673 (1972) (hereinafter *Benson*) before the Supreme Court's decision in *Parker* v. *Flook*, 437 U.S. 584, 198 USPQ 193 (1978) (hereinafter *Flook*). In his final rejection, dated October 27, 1976, the examiner stated that the subject matter "deemed as

the invention" is "a data structure" and then made the following analysis:

> The term "data structure" as recited in the claim is comprised of four entities described as "first means", "second means", "third means", and "fourth means." Whether these so called "means" are hardware structure which store coded signals for performing a particular function or whether they refer to coded signals representing computer control words per se, any life which the claim may have in view of that which is admittedly old in the art, resides in the "technique" (page 5, line 4) or "switch-system base instruction" (page 5, lines 13 and 14) for solving the problem of changing information of the system base which is located in scratchpad registers which software cannot access (page 4, line 20). The invention then is embodied in the coded signals representing instructions to the computer, the running process word RPW (first means of claim), priority level indicator word PRI (second means), J Table word and G Table word (third means), and micro-instruction of Figures 15c [sic, 15b] and 15c as found in the central store 1301. The invention resides in a "data structure" or an algorithm designed to control the multiprogramming computer to solve the particular problem indicated.
>
> Under the ruling in *Gottschalk* v. *Benson*, 175 USPQ 673, the instant claims, depending upon a program implemented algorithm for patentability, are deemed nonstatutory subject matter.

Appellants requested reconsideration and argued that their claims are directed to data structures in

hardware which are "specific new, novel and unobvious [in] arrangement." They asserted that by stating that the invention resided in a "technique," the examiner was clearly disregarding the claims and interpreting the invention strictly on the basis of what is found in the specification, because no "technique" is *claimed.* They stated that even if a technique (i.e., process) were claimed, *Benson* does not render all such inventions nonstatutory, and that their invention does not involve a *mathematical* algorithm.[2]

In his Answer before the board, the examiner noted that all of the limitations found in claim 1 were old in the art and that the "claim is thus reciting prior art coupled with subject matter which the U.S. Supreme Court has found to be non-statutory in *Benson.*" Appellants responded in their reply brief that it makes no difference whether individual elements are old in the art and that it is the elements in combination which define the invention as a whole.

---

[2] In *Benson,* the Supreme Court was clearly limiting its discussion to mathematical algorithms:

> The patent sought is on a method of programming a general purpose digital computer to convert signals from binary coded decimal form into pure binary form. *A procedure for solving a given type of mathematical problem is known as an "algorithm." The procedures set forth in the present claims are of that kind; that is to say, they are a generalized formulation for programs to solve mathematical problems of converting one form of numerical representation to another.* [409 U.S. at 65, 175 USPQ at 674 (emphasis ours).]

It is not clear whether the examiner regarded appellants' invention as a mathematical algorithm, but, as will become evident, the board did so regard the invention.

The Board

The board rendered its decision on September 20, 1978, after the Supreme Court's decision in *Flook*. After incorrectly stating that none of the claims recited the term "firmware," [3] the board analyzed appellants' claims element by element, concluding that the only novel arrangement of the recited structures resided in the microprogramming, "which together with its attendant memory hardware appears to constitute firmware."

Apparently on the basis of *Flook*, the board affirmed the rejection because it was of the opinion that the appealed claims are directed to a method of calculation or mathematical algorithm. The board found the claims similar to the claims at issue in *In re Waldbaum*, 559 F.2d 611, 194 USPQ 465 (CCPA 1977), which included language characterized by the board as "obviously related to calculating and mathematical problem solving." [4] Although the claims

---

[3] Claim 3, as well as dependent claims 4-6, explicitly recites firmware. Firmware is a term of art in the computer field and refers to microinstructions permanently embodied in hardware elements. For a further discussion, *see generally* Ross, *The Patentability of Computer "Firmware,"* 59 JPOS 731 (1977). We need not and do not decide at this time whether firmware *per se* is statutory under 35 USC 101 because the invention *as a whole* is not directed thereto. See our further discussion, infra.

[4] Waldbaum claim 1 reads in pertinent part:

> a method * * * to *count the number* of busy lines * * * *comparing means* * * * *to derive the number of l's* in said data word * * *. [Emphasis ours.]

[Footnote continued on page 8a]

here at issue do not contain similar mathematical language, the board said that this "does not make the functions attendant the 'means' of appellants' claims any less mathematical or less related to an algorithm within the meaning assigned that term by the USSC in *Benson*."

To support its conclusion that the appealed claims are mathematical in nature, the board relied on a statement in the specification to the effect that all of the data in the computer are in binary form, but may be interpreted as binary coded decimal, decimal or alphanumeric. We reproduce the board's reasoning in full:

> Since digital computers normally operate in some numerical radix, binary, binary coded decimal, or the like, we consider the operation of appellants' claimed invention to be mathematical.
>
> Every operation performed in appellants' invention as claimed involves the accommodation of data and instructions to the size of the registers in memory, and to the positional assignment to the registers in memory by the use of some numerical measure or quantity effected by way of electrical signals. In whatever form the instructions employed in appellants' invention are characterized, numerical or otherwise, we think it is accurate to say that the operation of appellants'

---

[4] [Continued]

It is clear that this claim recites a mathematical algorithm. It solves a mathematical problem, to wit, *counting* a number of busy lines in a telephone system.

> structure is mathematical and that the instructions constitute a procedure which is algorithmic in character, to the same degree as that of the Waldbaum structure and that satisfactory operation of the apparatus claimed represents the successful solution of a mathematical problem falling within the definition of algorithm supplied in *Benson* and reiterated in *Flook*.
>
> Here the method of calculation is to be used in the internal operation of the computer system and the claims recite such an end use by means of the functions recited for the various means. A claim for an improved method of calculation, even when tied to a specific end use, is unpatentable subject matter under 35 USC 101, *Flook, supra.*

In summary, the board stated that the claims are drawn to apparatus in form only, and couple the apparatus (which it asserts is old in the art) "with subject matter, namely, programming, which is nonstatutory under the *Benson*, *Christensen*,[5] *Waldbaum* and *Flook* cases * * *."

OPINION

The examiner's basis for the rejection is grounded on the erroneous interpretation of the Supreme Court's decision and opinion in *Benson*, namely, that all computer program or program related inventions are nonstatutory under § 101. Both the Supreme

---

[5] *In re Christensen*, 478 F.2d 1392, 178 USPQ 35 (CCPA 1973).

Court [6] and this court [7] have thoroughly repudiated this view. Our decision, therefore, is based solely on the analysis made by the board.

The board said that the claims do not *directly* recite a mathematical formula, algorithm, or method of calculation, but, nevertheless, held the claims to be mathematical in nature. As appears from the quoted portion of the board opinion, the board regarded the fact that digital computers operate in some number radix as conclusive on the issue of whether the appealed claims recite a mathematical algorithm in the *Benson* and *Flook* sense. The board did not, however, direct attention to any specific formula it thought is utilized, or to what, if anything, the mathematical calculations alleged to be present in the claims are directed.

---

[6] In *Benson*, the Court stated, 409 U.S. at 71, 175 USPQ at 676 (emphasis ours):

> It is said that this decision precludes a patent for any program servicing a computer. *We do not so hold.*

In *Flook*, the Court stated, 437 U.S. at 595 n.18, 198 USPQ at 199 n.18:

> Very simply, our holding today is that a claim for an improved method of calculation, even when tied to a specific end use, is unpatentable subject matter under § 101.

[7] *See In re Gelnovatch*, 595 F.2d 32, 36-37, 201 USPQ 136, 141 (CCPA 1979); *In re Johnson*, 589 F.2d 1070, 1075, 200 USPQ 199, 205 (CCPA 1978); *In re Sarkar*, 588 F.2d 1330, 1333, 200 USPQ 132, 137 (CCPA 1978); *In re Freeman*, 573 F.2d 1237, 1244, 197 USPQ 464, 470 (CCPA 1978); *In re de Castelet*, 562 F.2d 1236, 1240, 195 USPQ 439, 443 (CCPA 1977); *In re Chatfield*, 545 F.2d 152, 155-56, 191 USPQ 730, 733-34 (CCPA 1976), *cert. denied*, 434 U.S. 875, 195 USPQ 465 (1977).

We do not agree with the board. We are constrained to reject its reasoning. Such reasoning leads to the conclusion that any computer-related invention must be regarded as mathematical in nature, a conclusion which is not compelled by either *Benson* or *Flook*.

The board's analysis confuses *what* the computer does with *how* it is done. It is of course true that a modern digital computer manipulates data, usually in binary form, by performing mathematical operations, such as addition, subtraction, multiplication, division, or bit shifting, on the data. But this is only *how* the computer does what it does. Of importance is the significance of the data and their manipulation in the real world, i.e., *what* the computer is doing. It may represent the solution of the Pythagorean theorem, or a complex vector equation describing the behavior of a rocket in flight, in which case the computer is performing a mathematical algorithm and solving an equation. This is what was involved in *Benson* and *Flook*. On the other hand, it may be that the data and the manipulations performed thereon by the computer, when viewed on the human level, represent the contents of a page of the Milwaukee telephone directory, or the text of a court opinion retrieved by a computerized law service. Such information is utterly devoid of mathematical significance. Thus, the board's analysis does nothing but provide a quick and automatic negative answer to the § 101 question simply because a computer program is involved.

Appellants have continuously insisted that they are claiming a new and unobvious combination of *hardware elements*, i.e., a new machine or apparatus.[8] The issues of novelty and unobviousness[9] are not before us, but we agree with appellants insofar as they characterize their invention as a machine or apparatus. The board likewise seems to agree on this point. In its opinion, it identifies all of the "means" of appellants' claim 1 as structural hardware elements, such as registers, portions of main memory and control store,[10] and other computer components. Thus, the claim falls literally within the boundaries of § 101.

Appellants have characterized their combination of hardware elements as a mechanism which enables the computer to alter information in its system base in a manner not previously possible. They are in no way claiming the altered information; in fact, the particular information acted upon by appellants' invention is irrelevant to the operation of the invention itself.

---

[8] 35 USC 101 states:

> Whoever invents or discovers any new and useful process, *machine*, manufacture, or composition of matter, *or any new and useful improvement thereof*, may obtain a patent therefor subject to the conditions and requirements of this title. [Emphasis ours.]

[9] 35 USC 102, 103.

[10] The control store of a computer is a plurality of multibit storage locations (memory) containing microinstructions, which, when decoded by appropriate circuitry, provide control signals which cause specific operations to take place in the computer's CPU.

We see no difference in this regard, with respect to being within § 101, between appellants' claimed invention and a strictly mechanical adding machine, which is certainly statutory if claimed in a manner which does not embrace any particular calculation that the machine is capable of making.

The PTO regards as significant the fact that firmware is involved in the present invention. In a sense, firmware may be likened to software (computer programs); it is information which has been embodied into hardware by, for example, destroying fusible links in a read only memory (ROM) array. In appellants' invention, the information contained within the firmware, which is located in the control store of the computer, directs the desired information transfers within the system base.

If appellants were claiming the information embodied in the firmware or the firmware itself, per se, a different case would be presented. We express no opinion on the statutory nature of such an invention, a question not before us. Appellants are claiming a combination of hardware elements, one of which happens to be a portion of the computer's control store microprogrammed in a particular manner. It is this subject matter with which we must deal.

From our reading of appellants' specification, their claimed "data structure" is merely that which results from the arrangement of the recited hardware elements in the claimed manner. It is the result of certain structural "means" performing certain recited functions as explicitly sanctioned by 35 USC

112, sixth paragraph. We disapprove the board's distillation of appellants' claim down to the information contained in the firmware in order to hold it nonstatutory. The invention is not claimed in that manner, and, in this case, we see no reason to view the claim format as a subterfuge for masking the presence of an essentially nonstatutory invention.

Even though the claimed invention is a machine, we must nevertheless determine whether the claim recites a mathematical algorithm, and, if so, whether it preempts the use of the algorithm. *In re Noll*, 545 F.2d 141, 148, 191 USPQ 721, 726 (CCPA 1976). In doing so, we apply the two-step test in *In re Freeman*, supra note 7.

When we examine appellants' invention as a whole under the first step of this test, including the information microprogrammed into the firmware element as depicted in Figs. 14(a-i) and 15(b-c), we fail to detect the presence of any *mathematical* algorithm. In altering information in the system base as desired, certain "calculations" are made, such as determining whether a given quantity is equal to zero, or, as noted by the solicitor, multiplying an address in memory by sixteen to arrive at another address. However, it certainly cannot be said that comparing with zero or multiplying by sixteen is preempted by appellants' claims. Furthermore, the presence of these calculations does not transform the invention as a whole into a method of calculation. *Cf. In re Gelnovatch*, supra note 7. There is no solution of an equation, such as the new alarm limit in *Flook*, or the equiva-

lent pure binary number, as in *Benson*, present in the computer after the task has been completed. *See In re Chatfield*, supra note 7.

In summary, we have examined the claims thoroughly and we do not find any mathematical formula or mathematical method of calculation, improved or otherwise, which is either claimed as such or essential to the vitality of the claims. We find that the invention is a combination of tangible hardware elements —a machine—including some hardware elements which contain microprogrammed information termed "firmware." We do not find the invention to be nonstatutory under the authority of *Benson* or *Flook*, or under the authority of our own cases, such as those cited at note 7 ante. Therefore, the decision of the board is *reversed.*

REVERSED

APPENDIX B

UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

MANDATE

No. 79-533

IN THE MATTER OF THE APPLICATION

OF

JOHN J. BRADLEY and BENJAMIN S. FRANKLIN

Serial No. 570,331

ON APPEAL from the Board of Appeals

This CAUSE having been heard and considered, it is

ORDERED and ADJUDGED: Reversed

Dated July 5, 1979

A True Copy
GEORGE E. HUTCHINSON,
Test:
Clerk, United States Court of
Customs and Patent Appeals

Certified this 27th day of July 1979

/s/ George E. Hutchinson
GEORGE E. HUTCHINSON

APPENDIX C

OPINION AND DECISION OF BOARD OF APPEALS, SEPTEMBER 20, 1978.

Before Burns and Spencer, Examiners-in-Chief, and Craig,

Acting Examiner-in-Chief.

Burns, Examiner-in-Chief

This is an appeal from the final rejection of claims 1-6, constituting all of the claims presently in the case.

The subject matter concerns computer "firmware." [1] In their brief appellants characterize their invention in the following words:

> ". . . More specifically, the invention concerns a combination of hardware including a model-independent instruction means which is implemented in firmware, for adding capability and increased functionality to the computer system. As defined by the Applicants' claims, the present invention is directed to a combination of hardware structures controlled by a microprogrammed control system which provides for the generation of signals which cause the creation of

---

[1] The term has been defined as follows:

firmware-Logic circuits in read-only memory that may be altered by the software under certain circumstances, (*Computer Dictionary and Handbook,* Sippl and Sippl, published by Howard W. Sams & Co., Inc., The Bobbs-Merrill Co. Inc., 2nd Ed., 1972.)

an image of a communication center at another location, and for the generation of signals which causes the interchange of the information between the commuication center and its image."

More particularly, the problem addressed and appellants' solution thereto may be discerned from the following selected portions of appellants' disclosure. From page 4, beginning at line 15:

"High performance computing systems find it necessary to store portions of the system base in internal scratchpad registers instead of main memory, and hence inaccessible and invisible to software running in the system. Accordingly, a problem results when software (i.e. the programmer) wants to change positions or change information of the system base which happens to be located in these internal central processor scrathpad [*sic*] registers which software cannot access."

From the same page, beginning at line 24:

"The problem has been solved in one of two ways. The first method is to simply re-initialize the entire system, loading the system base with different contents as a result of a new initialization process. This technique is extremely time-consuming and quite undesirable.

"A second technique for solving the problem is to have the software employ special model-independent properties of the particular machine on which it was running, to change the system base. This results in the very undesirable property of completely model-dependent software—

a feature which certainly does not serve to endear a computer manufacturer to a customer.

"What is needed therefore is a technique of changing information in the system base both in main memory and internal scratchpad registers making the system base in scratchpad memory accessible to software but not visible to software i.e. cannot be read by the software."

From page 5, beginning at line 26:

"A model-independent instruction in firmware is provided to change, upon its execution, information in the system base which is located in both main memory and internal scratchpad registers.

"The system base located at a location specified by the boundary address register BAR in main memory, is copied into a new location in main memory with predetermined portions of its information changed. This is readily accomplished by the operating system. . .

* * * * *

The switch system base instruction then transfers the information which is in the system base at the new location into the system base at the location specified by BAR and also that portion of the system base located in scratchpad memory."

We reproduce claim 1 as representative, however, in so doing, we interleave indented explanation (identified parenthetically with capital letters A to D) of the respective portions of the claimed subject matter, the explanation being derived from pages 9 and 10 of appellants' brief

1. In a multiprogramming computer system having a main memory, a central processing unit (CPU) coupled to said main memory, said (CPU) controlling the state of a plurality of groups of processes being in a running, ready, wait or suspended state, said computer system also having scratchpad registers being accessible to an operating system for controlling said multiprogramming computer system, a data structure for storing coded signals for communicating between said processes and said operating system, and said scratchpad registers, said data structure comprising:

(a) first means in said data structure and communicating with said operating system for storing signals indicative of an address for a selected one of said processes;

> (A) ". . . data structure claimed is comprised of the first means which is the structure of the running process word register at BAR + 56 of the system base structure of Figure 6. The system base structure is described with the particularity of 35 USC 112 beginning on page 35 through 41. Its use is further described beginning on page 41. The running process word register structure is further shown with particularity on Figure 11j and is described with particularity beginning on page 39 of the specification. . ."

(b) second means in said first means for storing coded signals indicating priority of said selected one

of said processes in relation to others of said processes for obtaining control of said CPU when ready;

> (B) ". . . The second means which is an element within the first is part of the running process word register and is identified with reference to numeral 1129. . ."

(c) third means in said data structure and communicating with said operating system, for storing coded signals indicative of an address for a selected one of said plurality of groups of processes; and,

> (C) ". . . The third means within the data structure is the J table word means and the G table word means at BAR and BAR + 4 on Figures 11a and 11b and described in the specification. . ."

(d) fourth means coupled to said data structure and said scratchpad registers, for generating signals causing the changing of information in said data structure and said scratchpad registers.

> (D) ". . . Finally, the fourth means claimed is shown on Figures 13a and 13b is a central store as microprogrammed by the microprogram as disclosed by the flow charts of Figures 15b and 15c. Details of the control store which is microprogrammed are shown on Figure 13b (it should be understood that control stores are old in the art and persons of ordinary skill in the art

can microprogram them when provided with the flow charts of the microprogram and this results in the firmware. . ."

THE REJECTION

All of the claims have been rejected under 35 USC 101 as drawn to non-statutory subject matter. The examiner cites *Gottschalk* v. *Benson*, 409 US 63, 175 USPQ 673 (1972).

In his answer, beginning at page 5, line 6, the examiner explains his position in part as follows:

> ". . . Claim 1 has been couched in apparatus format and recites first means through fourth means. The first means through third means set forth means for storing signals and read upon registers or memory locations within the operating system, already shown to be old in the art. The function of or attribute of the signals stored in the first, second, and third means for storing may be classified as indicating an address for a selected one of a plurality of processes, indicating a priority of said selected process relative to other processes, and indicating an address of a selected one of the plurality of processes, respectively. There is clearly no novelty to be found here for distinguishing over that which is deemed to be old in the art.
>
> "The fourth means is stated as means for generating signals causing the changing of information in the data structure and scratchpad registers and is categorized by applicants' brief, page 10, as a control store as microprogrammed by the microprogram of Figures 15b and 15c.

> The use of control stores in computer systems has been shown to be old in the art in the listing of publications and U.S. Patents in paragraph 19 of Appendix I, amendment filed August 11, 1976. In fact, paragraph 22 of the Appendix I correctly indicates that microprogrammed computer processors including microprogrammed control units are old in the art and suggests that the systems shown in the two cited patents are capable of practicing or implementing the instant invention. . .
>
> * * * * *
>
> Applicants' invention resides in an instruction in firmware which changes, upon execution, information in the system base. The claim is thus reciting prior art coupled with subject matter which the U.S. Supreme Court has found to be non-statutory in *Benson*. . ."

The appellants' argument is to the effect that only hardware, that is, structure, has been recited wherefore the *Benson* decision is inapposite. They characterize the examiner's rationale as procrustean "in forcing an arbitrary conclusion or solution to the facts," brief, bottom of page 8.

OPINION

In order to determine what appellants regard as their invention, it is necesary to consider the claimed subject matter as a whole, *In re* Noll, 545 F.2d 141, 191 USPQ 721 (CCPA, 1976), *In re* Chatfield, 545 F.2d 152, 191 USPQ 721 (CCPA, 1976). With this caveat in mind it is nevertheless appropriate to con-

sider the features, both novel and otherwise, within the claims, as did the Court in *Parker* v. *Flook*, 437 US—, 198 USPQ 193 (1978), an approach "not at all inconsistent with the view that a patent claim must be considered as a whole."

Analysis of Claimed Subject Matter

First, we observe that neither the term "firmware" nor an apparent equivalent appears in any of the claims, cf. USC 112, last sentence. The various recited "means" for storing, communicating and generating are not to be construed as limited to firmware.

In an amendment filed August 11, 1976 (Paper No. 4), appellants amended their specification with APPENDIX I providing numbered items which refer to prior art features, to which APPENDIX we will have occasion to refer.

Our comments on the elements recited in the claims will follow their order of recitation within each claim.

Claim 1

For the main memory, appellants in APPENDIX I, (11)(a) appear to rely for disclosure upon references 2-7 found on page 2 of the specification. References 2-6 upon inspection appear to show a mix of block diagrams directed to both hardware and software. Reference 7 shows only flow charts without a suggestion of hardware or apparatus structure. To the extent that references 2-6 disclose structure or hardware it would appear that, as such, these references demonstrate that memory *per se* is old. It does

not appear that appellants have argued otherwise, rather, appellants urge that they claim ". . . new, novel and unobvious *arrangement* of these structures . . .," brief, page 9, beginning at line 4, (our emphasis).

As for the central processing unit (CPU), no specific reference has been identified in APPENDIX I by this term, nevertheless, item (22) thereof describes microprogrammed computer processors employing microprogrammed control units "with which the instant invention may be practiced."

Concerning the "data structure" of the claim, page 11, beginning at line 18, of the specification characterizes this feature as being composed of data formats, software visible registers, and instruction formats. The first and last of these three features are described in paragraphs A of specification page 11, line 21 and J of page 14, line 15. Nothing in this description of formats suggests hardware. It would seem that reliance must be placed on the software visible registers, the second feature, as described in paragraph E, page 12, beginning at line 24 of the specification. Appellants appear to rely as prior art for such features upon patents described in (12) and (13) of APPENDIX I.

Appellants have characterized the "first means" as the running process word register within the data structure. Save for the actual word stored which, it would appear, will change depending upon the nature of the data processing undertaken, this feature con-

sists of space in memory (BAR + 56 as shown in Figure 6) as part of the system base.

From the foregoing it would seem that the data structure consists of at least one software visible register, and the first means in said data structure constitutes that register (or a portion of it) with a particular word stored therein, an address. It does not appear that the "first means" constitutes novel hardware distinct from the registers designated above representing the data structure for which appellants have referenced prior art.

Similarly, for the second and third means of the claim, so much thereof as may be characterized as hardware would seem to have antecedents in the prior art registers as part of memory described above for the data structure.

The fourth means of the claim has been designated by the appellants as a central store, known to be old, brief, page 10, line 8. It is there characterized:

> ". . . as microprogrammed by the microprogram as disclosed by the flow charts of figures 15b and 15c. . ."

Appellants have buttressed this by the prior art references in APPENDIX I (19), (20) and (21).

From the above we conclude that the components recited in claim 1 appear to be old in the art constituted as registers as part of memory (the latter being hardware) save for the microprogramming provided by element (d) of the claim, the fourth means. Any "specific new, novel and unobvious arrangement of these structures," brief, page 9, beginning at line 4,

would appear to be provided by the microgramming of Figures 15b and 15c, which together with its attendant memory hardware appears to constitute firmware.

Claims 2-6

As to these claims, analysis of the nature set forth above in connection with claim 1 leads to a similar conclusion. We would only add that, with respect to claim 4, appellants have made reference to arithmetic and logic units in APPENDIX I(2)(a). The element characterized as "hardware gate" found in claim 5 appears also to be part of memory, for which appellants have provided prior art references, under the control of the programming shown in Figures 15b and 15c, see specification page 64, beginning at line 26 et seq., of the specification.

Authorities

Appellants in their brief rely upon *In re* Noll, *supra*, and *In re* Chatfield, 545 F.2d 152, 191 USPQ 730 (CCPA, 1976). In their reply brief, appellants comment upon *In re* Waldbaum, 559 F.2d 611, 194 USPQ 465 (CCPA, 1977) and urge the significance of the court's language in distinguishing the Waldbaum subject matter from that in *In re* Christensen, 478 F.2d 1392, 178 USPQ 35 (1973). Appellants there maintain that "the instant invention does not involve nor does it claim a method, process or algorithm directed to a calculation." We disagree. It appears to us that Waldbaum, if anything, supports the examiner's position, in that the Court proscribed

claims "which are directed to methods for controlling the operation of a data processor," basing its decision upon the *Benson* decision. It would seem that appellants' claims are similarly directed.

We recognize that method claims were under consideration in Waldbaum whereas here the claims are in apparatus format. In *In re* Freeman, 197 USPQ 464 (CCPA, 1978) the Court found such a circumstance to be of no moment:

> "Though a claim expressed in 'means for' (functional) terms is said to be an apparatus claim, the subject matter as a whole of that claim may be indistinguishable from that of a method claim drawn to the steps performed by the 'means.'
>
> "We agree with the solicitor's contention that if allowance of a method claim is proscribed by Benson, it would be anomalous to grant a claim to apparatus encompassing any and every 'means for' practicing that very method."

We are aware that the Waldbaum claims included characteristic language obviously related to calculating and mathematical problem solving, thus (from *Waldbaum* claim 1):

> ". . . a method . . . to *count* the number of busy lines. . . . . *comparing* means . . . to derive the *number of 1's* in said data word." (Emphasis added.)

That appellants' claim are absent similar characteristic language does not make the functions attendant the "means" of appellants' claims any less mathe-

matical or less related to an algorithm within the meaning assigned that term by the USSC in *Benson*.

The specification states, page 12, line 2:

> "All data are in binary form, but may be interpreted as binary, decimal or alphanumeric."

Since digital computers normally operate in some numerical radix, binary, binary coded decimal, or the like, we consider the operation of appellants' claimed invention to be mathematical.

Every operation performed in appellants' invention as claimed involves the accommodation of data and instructions to the size of the registers in memory, and to the positional assignment to the registers in memory by the use of some numerical measure or quantity effected by way of electrical signals. In whatever form the instructions employed in appellants' invention are characterized, numerical or otherwise, we think it is accurate to say that the operation of appellants' structure is mathematical and that the instructions constitute a procedure which is algorithmic in character, to the same degree as that of the Waldbaum structure and that satisfactory operation of the apparatus claimed represents the successful solution of a mathematical problem falling within the definition of algorithm supplied in *Benson* and reiterated in *Flook*.

Here the method of calculation is to be used in the internal operation of the computer system and the claims recites such an end use by means of the functions recited for the various means. A claim for an

improved method of calculation, even when tied to a specific end use, is unpatentable subject matter under 35 USC 101, *Flook, supra.*

SUMMARY

We agree with the examiner. The claims under consideration, when each is considered as a whole, recite apparatus as a matter of form only, and couple such apparatus, which appellants have indicated as old in the art, with subject matter, namely, programming, which is nonstatutory under the *Benson, Christensen, Waldbaum* and *Flook* cases, *supra.* Employing appellants' metaphor, the claims at bar, we think, extend beyond the procrustean statutory bed.

The decision of the examiner is affirmed.

AFFIRMED

/s/ Burns
Examiner-in-Chief

/s/ Spencer
Examiner-in-Chief

/s/ Craig
Examiner-in-Chief
(Acting)

BOARD OF APPEALS

APPENDIX D

LETTER OF EXAMINER, MARCH 12, 1976.

☒ This application has been examined

Responsive to communication filed ☐ This action is made final.

A SHORTENED STATUTORY PERIOD FOR RESPONSE TO THIS ACTION IS SET TO EXPIRE THREE (3) MONTH(S) FROM THE DATE OF THIS LETTER.

PART I

The following attachments(s) are part of this action:

a. ☒ Notice of References Cited, Form PTO-892. b. ☐ Notice of Informal Patent Drawing, PTO-948.

c. ☐ Notice of Informal Patent Application, Form PTO-152. d. ☐

PART II

Summary of Action

1. ☒ Claims 1 - 6 are presented for examination.

2. ☐ Claims _____ are allowed.

3. ☐ Claims _____ would be allowable if amended as indicated.

4. ☒ Claims 1 - 6 are rejected.

5. ☐ Claims _____ are objected to.

6. ☐ Claims _____ are subject to restriction or election requirement.

7. ☐ Claims _____ are withdrawn from consideration.

8. ☐ Since this application appears to be in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under Ex parte Quayle, 1935 C.D. 11; 453 OG. 213.

9. ☐ Since it appears that a discussion with applicant's representative may result in agreements whereby the application may be placed in condition for allowance, the examiner will telephone the representative within about 2 weeks from the date of this letter.

10. ☐ Receipt is acknowledged of papers under 35 USC 119, which papers have been placed of record in the file.

11. ☐ Applicant's claim for priority based on an application filed in _____ on _____ is acknowledged. It is noted, however, that a certified copy as required by 35 USC 119 has not been received.

12. ☐ Other

1) Due to the length of the specification it has not been checked to the extent of determining all minor errors. Applicants are requested to review the entire specification and correct any errors noted.

2) The Abstract of the Disclosure should be reviewed and corrected to more specifically and clearly state the character of the subject matter deemed to be new in the art.

3) In the specification, pages 2-4 and various others throughout the disclosure, application serial numbers and filing dates have been omitted.

In lines 1 and 2, page 2, the purpose of the procedure to allow incorporation by reference to previous patents and applications was to reduce the length of disclosure in related applications and simplify the examining and printing procedures involved. Applicants have incorporated by reference numerous applications and/or patents which do not appear to be "essential subject matter" relative to the disclosure of the instant invention. If these prior applications are to be referenced as related applications then it should be so indicated rather than incorporating all of them by reference.

4) The specification is objected to under 35 USC 112 as failing to set forth the subject matter of the invention in such terms as to enable one to make and use the invention without undue experimentation. The basic elements of the system are disclosed in no more detail than hollow rectangles and the complex interrelationship between the various elements is disclosed only in functional statements of desired re-

sults. For example, the control unit 130 of Figure 1 is stated to contain firmware which initializes the system, controls the CPU 104 and IOC 105, and encodes an instruction set. However, the elements of the control unit and the instructions for configuring the various elements have not been disclosed in sufficient detail to enable one of ordinary skill in the art to reconstruct the invention.

Applicants have referred to basic data structures on page 11 and following disclosure as being data formats, software visible registers, and instruction formats. This use of the term "structures" to refer to data formats, etc. is carried over into the claims and results in confusion as to the nature of the invention. The subject matter disclosed in Figure 2 must be deemed in the nature of data format and not of the nature of apparatus as usually intended by use of the word "structure." Clarification is requested.

5) Claims 1-6 are rejected as drawn to subject matter which is inadequately disclosed, 35 USC 112, first paragraph.

6) Claims 1-6 are further rejected under 35 USC 112, second paragraph, as failing to particularly point out and distinctly claim the subject matter sought to be patented.

The structural connections and cooperative relationship between the various elements comprising the claimed combination have not been clearly established. Although the claims are recited as a combination of devices in the means plus statement of func-

tion language, most of the claim language does not recite "structure" in the sense of apparatus but rather uses "means" to refer to software or firmware instructions. Applicant is therefore requested to point out the structure in the drawings upon which the various "means" are to be read.

7) Claims 1-6 are further rejected under 35 USC 101 as drawn to non-statutory subject matter.

The Supreme Court in *Gottschalk* v. *Benson*, 175 USPQ 673, stated that Court's views on the patentability of certain claims which set forth an invention related to the processing of data by a programmed computer.

The Court in Benson addressed itself to the broad question of patentability of computer programs and stated that if such subject matter is to be patented, considerable programs are raised which only committees of Congress can manage. Justice Douglas, speaking for the Court, declined to decide such policy matters and refused to extend the patent statutes to embrace computer programming, absent considered action by the Congress.

8) Claims 1-6 are further rejected under 35 USC 102 as unpatentable over the Hauck patent. The apparatus in Figure 1 of Hauck cooperates to perform the function of applicant's claimed invention.

9) The additional art cited, although not applied against the claims, is considered pertinent in showing related systems.

10) All claims have been rejected. This application should be put in condition for final disposition.

/s/ Charles E. Atkinson
CHARLES E. ATKINSON
EXAMINER
GROUP ART UNIT 236

LETTER OF EXAMINER, OCTOBER 27, 1976.

[x] Responsive to communication filed on August 11, 1976.

[x] This action is made final.

A SHORTENED STATUTORY PERIOD FOR RESPONSE TO THIS ACTION IS SET TO EXPIRE THREE (3) MONTH(S) ~~DAYS~~ FROM THE DATE OF THIS LETTER.

FAILURE TO RESPOND WITHIN THE PERIOD FOR RESPONSE WILL CAUSE THE APPLICATION TO BECOME ABANDONED. 35 U.S.C. 133

PART I THE FOLLOWING ATTACHMENT(S) ARE PART OF THIS ACTION:

1. [ ] Notice of References Cited, Form PTO-892.
2. [ ] Notice of Informal Patent Drawing, PTO-948.
3. [ ] Notice of Informal Patent Application, Form PTO-152
4. [ ]

PART II SUMMARY OF ACTION

1. [x] Claims 1 - 6 are pending in the application.

   Of the above, claims _____ are withdrawn from consideration.

2. [ ] Claims _____ have been cancelled.

3. [ ] Claims _____ are allowed.

4. [x] Claims 1 - 6 are rejected.

5. [ ] Claims _____ are objected to.

6. [ ] Claims _____ are subject to restriction or election requirement.

7. [ ] The formal drawings filed on _____ are acceptable.

8. [ ] The drawing correction request filed on _____ has been [ ] approved. [ ] disapproved.

9. [ ] Acknowledgement is made of the claim for priority under 35 U.S.C. 119. The certified copy has
   [ ] been received. [ ] not been received. [ ] been filed in parent application; serial no. _____ filed on _____

10. [ ] Since this application appears to be in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under Ex parte Quayle, 1935 C.D. 11; 453 OG. 213.

11. [ ] Other

1) This is in response to the amendment filed August 11, 1976. Claims 1 through 6 remain in the application.

2) Claims 1 through 6 are again rejected as being directed to non-statutory subject matter under 35 USC 100 and 101.

Applicants contend that "the present invention is directed to hardware structures controlled by a microprogrammed control system which provides for the generation of signals which cause the creation of an image of a communication center at another location and for the generation of signals which causes the interchange of the information between the communication center and its image." Applicants further contend that "the structure and microprogram control system defined by applicants' claims are clearly within the technological arts" and is thus statutory subject matter within the intended bounds of 35 USC 100 and 35 USC 101.

Although applicants have now amended the claims to couch the claimed subject matter in language inferring structural entities and interconnections, the disclosed invention and specification language clearly indicate that the invention resides in a "technique" or "switch base instruction" for changing information in the system base in both the main memory and internal scratchpad registers making the system base in scratchpad memory accessible to software but not visible to software i.e. cannot be read by the software.

In claim 1, the preamble sets forth the environment as a multiprogramming computer system comprised of such well known components as a main memory, central processing unit, and scratchpad registers. The subject matter deemed as the invention is then stated as "a data structure." The nature of this subject matter is disclosed beginning on page 35 of the specification and in Figure 6. Beginning in line 12 of page 35, it is stated that "the format of the system base 600 is shown." It is further stated that the "system base resides in absolute main memory and is developed by firmware." The term "data structure" as recited in the claim is comprised of four entities described as "first means", "second means", "third means", and "fourth means." Whether these so called "means" are hardware structure which store coded signals for performing a particular function or whether they refer to coded signals representing computer control words per se, any life which the claim may have in view of that which is admittedly old in the art, resides in the "technique" (page 5, line 4) or "switch-system base instruction" (page 5, lines 13 and 14) for solving the problem of changing information of the system base which is located in scratchpad registers which software cannot access (page 4, line 20). The invention then is embodied in the coded signals representing instructions to the computer, the running process word RPW (first means of claim), priority level indicator word PRI (second means), J Table word and G Table word (third means), and micro-instructions of Fig-

ures 15c and 15c as found in the central store 1301. The invention resides in a "data structure" or an algorithm designed to control the multiprogramming computer to solve the particular problem indicated.

Under the ruling in *Gottschalk* v. *Benson*, 175 USPQ 673, the instant claims, depending upon a program implemented algorithm for patentability, are deemed nonstatutory subject matter.

3) All claims have been rejected. This action is hereby made a *FINAL REJECTION*.

CHARLES E. ATKINSON
EXAMINER
GROUP ART UNIT 236

LETTER OF EXAMINER, FEBRUARY 1, 1977.

ADVISORY ACTION

☒ THE PERIOD FOR RESPONSE IS EXTENDED TO RUN 4 MONTHS FROM THE DATE OF THE FINAL REJECTION. 855 O. G. 1109.

☐ Appellant's Brief is due in accordance with Rule 192 (a).

Applicant's response to the final rejection, filed Dec. 30, 1976, has been considered with the following effect, but it is not deemed to place the application in condition for allowance:

1. ☐ The proposed amendments to the claims and/or specification will not be entered and the final rejection stands because:

   a. ☐ There is no convincing showing under Rule 116 (b).
   b. ☐ They raise new issues that would require further consideration and/or search.
   c. ☐ They raise the issue of new matter.
   d. ☐ They are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal.
   e. ☐ They present additional claims without cancelling a corresponding number of finally rejected claims.

2. ☐ Newly proposed or amended claims _____ would be allowed if submitted in a separately filed amendment cancelling the non-allowable claims.

3. ☒ Upon the filing of an appeal, the proposed amendment ☒ will be ☐ will not be, entered and the status of the claims in this application would be as follows:

   a. ☐ Claims _____ would be allowable.

   b. ☒ Claims 1 - 6 would not be allowable.

   However: (See attachment for further explanation.)

   (1) ☐ The rejection of claims _____ on references is deemed to be overcome by applicant's response.

   (2) ☐ The rejection of claims _____ on non-reference grounds only is deemed to be overcome by applicant's response.

4. ☐ The affidavit, exhibit or request for reconsideration has been entered but does not overcome the rejection.

5. ☐ The affidavit or exhibit will not be admitted because applicant has not shown good and sufficient reasons why it was not earlier presented.

6. ☐ The application having been examined under the special accelerated examining procedure (M. P. E. P. 708.02), the proposed amendment has not been considered since it does not prima facie place the application in condition for allowance or in better condition for appeal.

Claims 1-6 are deemed properly rejected under 35 USC 101 as drawn to non-statutory subject matter. The U.S. Supreme Court in rendering a decision on the *Dann* v. *Johnston* case, 189 USPQ 257, held the invention to be obvious under 35 USC 103, thus reversing the CCPA. The Court did not render a decision on the 35 USC 101 issue and moreover did not modify its position taken in the *Benson* v. *Gottschalk* case.

In both the *In re Noll* and *In re Chatfield* cases the CCPA reversed the Board of Appeals decision. Until these decisions become final and since further review of clarification may be forthcoming, the PTO is continuing to examine computer program related applications in accordance with its current interpretation of the Supreme Court decision in the *Benson* case.

/s/ Charles E. Atkinson
CHARLES E. ATKINSON
EXAMINER
GROUP ART UNIT 236

APPENDIX E

APPLICATION OF JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN, FILED APRIL 21, 1975, SERIAL NUMBER 570,331, FOR SWITCH SYSTEM BASE MECHANISM.

ABSTRACT OF THE DISCLOSURE

A model-independent instruction in firmware is provided to change, upon execution, information in the system base which is located in both main memory and internal scratch pad registers.

RELATED APPLICATIONS

The following applications are incorporated by reference to the instant application.

* * * * *

15. "System Base", invented by ————————, filed on ————————, and having Serial Number ———— and assigned to the same assignee named herein.

* * * * *

SUMMARY OF THE INVENTION

A model-independent instruction in firmware is provided to change, upon its execution, information in the system base which is located in both main memory and internal scratchpad registers.

The system base located at a location specified by the boundary address register BAR in main memory, is copied into a new location in main memory with predetermined portions of its information changed. This is readily accomplished by the operat-

ing system. (See application incorporated by reference into the application herein). The switch system base instruction then transfers the information which is in the system base at the new location into the system base at the location specified by BAR and also that portion of the system base located in scratch-pad memory.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features which are characteristic of the invention are set forth with particularity in the appended claims. The invention itself, however, both as to organization and operation together with further objects and advantages thereof may best be understood by references to the following description taken in conjunction with the drawings in which:

Figure 1 is a block diagram of a multiprogramming system utilizing the invention.

Figure 2 is a schematic representation of various hardware structures utilized by the invention.

Figure 3 is a legend of terms used for reserved areas of storage in registers depicted in Figure 2.

Figure 4 is a schematic diagram of a process control block.

Figure 5 is a schematic diagram of a system for addressing a process control block.

Figure 6 is a schematic diagram of the system base of the invention.

Figure 7A and 7B are a schematic representation of a stack segment and a stack frame respectively.

Figure 8 is a schematic diagram of a system for

addressing G-segments and in particular the queue of processes in the GO segment.

Figure 9 is an exploded schematic diagram of a GO segment illustrating queue of processes and process linking.

Figures 10a through 10L are block diagrams of structures in the PCB.

Figures 11a through 11r are block diagrams of structures in the system base.

Figure 12 is a schematic diagram of addressing schemes of user and system segments utilizing the system base and PCB structures.

Figure 13 is a schematic diagram of the control unit.

Figures 14a through 14i are flow diagrams of the dispatcher unit in firmware.

Figure 15a is a structure of the switch system base instruction.

Figures 15b-15c are flow diagrams of the switch system base instruction in firmware.

DESCRIPTION OF A PREFERRED EMBODIMENT

* * * * *

Control Unit

Referring to Figures 13a-13c details of the control unit are shown. The control unit, although shown separate from the central processing unit (CPU), is in actuality a part of the CPU and is comprised of a control store unit CSU 1301, a control store interface adapter CIA 1302 and appurtenant subunits,

control store loader CSL 1303 and control and load unit CLU 1304.

The control store unit CSU 1301 receives micro-instructions from the control store loader CSL 1303 via the control and load unit CLU 1304 and the control store interface adapter CIA 1302. Under normal operating conditions, microprograms are loaded from an external source during system initialization and become a permanent control function of the machine. However the control store unit CSU 1301 has the ability to be reloaded and initialized in a manner that provides for a variety of central processing unit CPU 1306 operational modes. The following modes of operation of the CPU are available under control of the CSU 1301; (a) native mode; (b) emulation mode; (c) concurrent native and emulation modes; (d) diagnostic mode. This capability is possible because the micro-instructions resident in the CSU are the source of micro-operations used to control the operation of all other CPU functional units such as the emulation unit 1316, the arithmetic logic unit ALU 1317, the instruction fetch unit IFU 1318, the address control unit ACU 1319 and the data management unit DMU 1321. Also shown within the central processing unit CPU 1306 are previously described general register 1307, base registers 1308, scientific registers 1309, T-registers 1310, status registers 1311, instruction counter IC 1312, and hardware control mask register 1313.

Typically the control store unit CSU 1301 is a 9K bipolar integrated circuit programmable read-only

memory (PROM) mixed with read/write random access store (RAM). It has a typical 150 nanosecond read cycle and a 450 nanosecond write cycle. Each location of control store stores one 84-bit micro-instruction word (to be later more fully described), and each micro-instruction word controls one CPU cycle. As each location of the control store of the control store unit CSU 1301 is read, its contents are decoded by micro-operation decoders which provide micro-operation control signals each of which causes a specific operation within the CPU to take place (to be later described in detail).

* * * *

AMENDMENT, AUGUST 11, 1976.

August 10, 1976

Commissioner of Patents & Trademarks
Washington
District of Columbia 20231

Dear Sir:

Responsive to the Office Action of 3/12/76 the due date of which was extended two months from 6/12/76 to and including 8/12/76, please amend the above-identified application as follows:

IN THE SPECIFICATION

* * * *

Page 3, line 21, after "invented by" insert—Claude Carre—.

Page 3, line 23, before ", and having" insert—December 2, 1974—.

Page 3, line 22, after "Number" insert—528,951—.

* * * *

Respectfully submitted,

/s/ Nicholas Prasinos
NICHOLAS PRASINOS
Registration No. 24,791
Attorney for Applicants

PRINT OF DRAWING
AS ORIGINALLY FILED

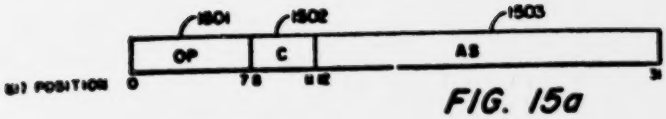


FIG. 15a

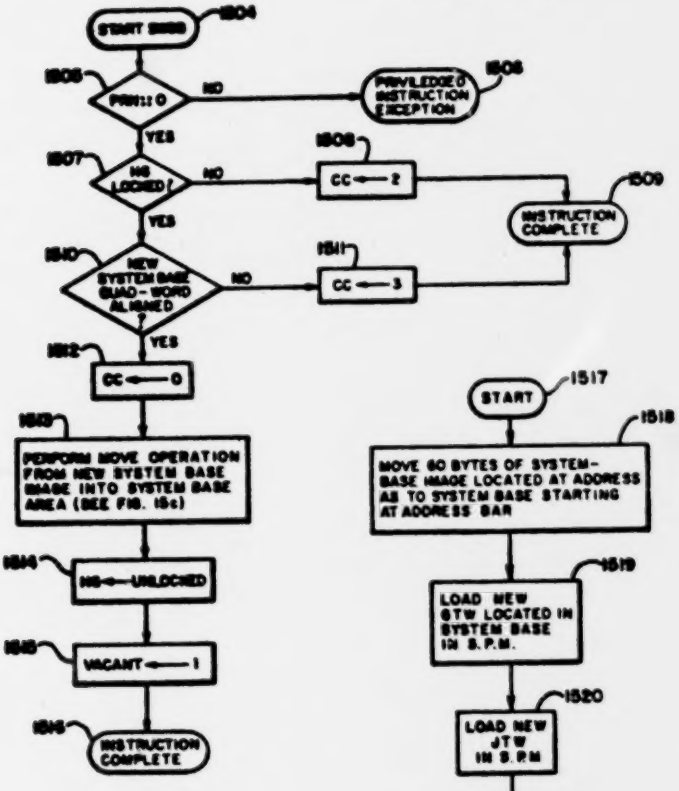


FIG. 15b

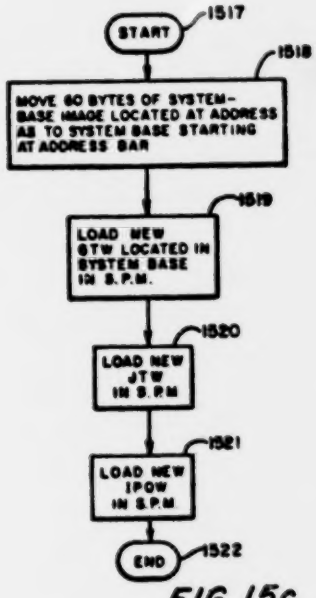


FIG. 15c

REJECTED CLAIMS.

1. In a multiprogramming computer system having a main memory, a central processing unit (CPU) coupled to said main memory, said (CPU) controlling the state of a plurality of groups of processes being in a running, ready, wait or suspended state, said computer system also having scratchpad registers being accessible to an operating system for controlling said multiprogramming computer system, a data structure for storing coded signals for communicating between said processes and said operating system, and said scratchpad registers, said data structure comprising:

(a) first means in said data structure and communicating with said operating system for storing coded signals indicative of an address for a selected one of said processes;

(b) second means in said first means for storing coded signals indicating priority of said selected one of said processes in relation to others of said processes for obtaining control of said CPU when ready;

(c) third means in said data structure and communicating with said operating system, for storing coded signals indicative of an address for a selected one of said plurality of groups of processes, and,

(d) fourth means coupled to said data structure and said scratchpad registers, for generating signals causing the changing of information in said date structure and said scratchpad registers.

2. In combination with a communication center located in a predetermined area of a computer main

memory the computer having a central processing unit (CPU) for executing plurality of processes that compete with each other for control of said CPU, operating system means coupled for providing information to said communication center, and a scratch pad memory in said CPU which stores, under control of said operating system means, portions of information located in said communication center, an improved system controlled by one of said processes for changing information in said communication area and in said scratchpad memory, said improvement comprising:

(a) first means for generating signals causing the creation of an image of said communication center at another location in said main memory;

(b) second means responsive to said operating system means for generating signals causing a change of the information in said image, resulting in a new image; and,

(s) third means, coupled for being responsive to said first and second means and to one of said processes for generating signals for causing a transfer of the new image to said communication center and to said scratchpad memory.

3. The system as recited in Claim 2 wherein said means is a switch system base instruction in firmware.

4. The system as recited in Claim 3 further including an arithmetic and logic unit having an auxiliary memory said arithmetic and logic unit coupled

for communicating with said first, second and third means.

5. The system as recited in Claim 4 wherein said auxiliary memory includes a hardware gate which has a locked and unlocked state said unlocked state permitting internal interrupts or external exceptions, and said locked state not permitting internal interrupts or external exceptions.

6. The system as recited in Claim 5 wherein said hardware gate is locked by said switch system base instruction when executing and unlocked by said switch system base instruction when it terminates execution.

APPENDIX F

APPLICATION OF CLAUDE CARRE, ET AL., FILED DECEMBER 2, 1974, SERIAL NUMBER 528951, FOR SYSTEM BASE

ABSTRACT OF THE DISCLOSURE

A system base provided a communication channel and interface and a mapping mechanism between the operating system and the firmware/hardware controlled processes in a multiprogramming/multiprocessing environment is disclosed. Processes in the multiprogramming/multiprocessing computer system may be in the running, ready, wait and suspended states. Whereas hardware/firmware control the running, ready, and wait states, the operating system creates and deletes processes within the multiprogramming/multiprocessing computer system. Accordingly a communication system is provided to pass information between operating system and processes and to start and stop related processes efficiently. The system base together with a boundary address register provide an interface and control center for such communication system.

* * * *

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features which are characteristic of the invention are set forth with particularity in the appended claims. The invention itself, however, both as to organization and operation together with further objects and advantages thereof may best be understood by references to the following description taken in conjunction with the drawings in which:

Figure 1 is a block diagram of a multiprogramming system utilizing the invention.

Figure 2 is a schematic representation of various hardware structures utilized by the invention.

Figure 3 is a legend of terms used for reserved areas of storage in registers depicted in Figure 2.

Figure 4 is a schematic diagram of a process control block.

Figure 5 is a schematic diagram of a system for addressing a process control block.

Figure 6 is a schematic diagram of the system base of the invention.

Figure 7A and 7B are a schematic representation of a stack segment and a stack frame respestively.

Figure 8 is a schematic diagram of a system for addressing G-segments and in particular the queue of processes in the GO segment.

Figure 9 is an exploded schematic diagram of a GO segment illustrating queue of processes and process linking.

Figures 10a through 101 are block diagrams of structures in the PCB.

Figures 11a through 11r are block diagrams of structures in the system base.

Figure 12 is a schematic diagram of addressing schemes of user and system segments utilizing the system base and PCB structures.

Figure 13 is a schematic diagram of the control unit.

Figures 14a through 14i are flow diagrams of the dispatcher unit in firmware.

* * * *

[SEAL]

U.S. DEPARTMENT OF COMMERCE
Patent Office

Address Only: COMMISSIONER OF PATENTS
Washington, D.C. 20231

Please Refer To The Following:

Group Art Unit—237
Applicant(s)—Claude Carre et al
Serial Number—528951 Filing Date 12/2/74
Title—System Base

Fred Jacob
Honeywell Info. Systems Inc.
200 Smith St.
Waltham, Mass. 02154

[Mailed Aug. 17, 1978, Group 230]

NOTIFICATION RE APPEAL

* * * *

2. ☒ The appeal in this application is DISMISSED because:
   A. ☐ The $50 fee for filing the Brief as required by 35 U.S.C. 41(a)(6) was not timely filed.
   B. ☒ The Brief was not filed, or was not timely filed (Rule 192(b)).
3. ☒ As the result of the dismissal in "2" above, this application:
   A. ☒ Is abandoned since there are no allowed claims.

* * * *

☆ U. S. GOVERNMENT PRINTING OFFICE: 1979 305885 142

Supreme Court, U.S.
FILED
MAY 30 1980
MICHAEL RODAK, JR., CLERK

APPENDIX

In the Supreme Court of the United States

OCTOBER TERM, 1979

No. 79-855

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS and TRADEMARKS,

Petitioner

—*v.*—

JOHN J. BRADLEY and BENJAMIN S. FRANKLIN

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

PETITION FOR A WRIT OF CERTIORARI FILED DECEMBER 3, 1979
CERTIORARI GRANTED MARCH 17, 1980

In the Supreme Court of the United States

OCTOBER TERM, 1979

No. 79-855

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS and TRADEMARKS,

Petitioner

—*v.*—

JOHN J. BRADLEY and BENJAMIN S. FRANKLIN

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

INDEX

	Page
Docket Entries	1
Letter of Transcriber dated December 7, 1979	3
Photographs	4
(a) Punch cards of the type utilized by programmers in computer programming	4
(b) Magnetic Tape of the type utilized by programmers in computer programming	5
(c) A computer hardwired board of the type utilized by system designers to design new computer machines	6
(d) Two computer firmware boards of the type utilized by system designers to design new computer machines	
Board (1)	7
Board (2)	8
(e) A MOS memory board of the type utilized for main memories in computer machines	9
Opinion of the United States Court of Customs and Patent Appeals dated July 5, 1979	10
Judgment of the United States Court of Customs and Patent Appeals dated July 27, 1979	20a

	Page
Praecipe for transcript of record dated November 17, 1978	21
Application Serial No. 570,331 (including petition, specification, drawings and declaration as filed, but not including claims)	24
Letter of the Examiner (First Office Action) March 12, 1976	121
Notice of references cited	125
Amendment, August 11, 1976	126
Letter of Examiner (Second Office Action) dated October 27, 1976	150
Letter to Office, December 30, 1976. Amendment under rule 116	153
Letter of Examiner, February 1, 1977. Advisory Action	163
Brief for Appellants, April 19, 1977	165
Examiner's Answer, August 4, 1977	179
Reply brief for Appellants, August 17, 1977	184
Opinion and Decision of Board of Appeals, September 20, 1978	193
Notice and Reasons of Appeal to United States Court of Customs and Patent Appeals, November 17, 1978	204
Rejected Claims	206
Order Granting Certiorari	208

DOCKET ENTRIES

Docket No. 79-533

IN THE MATTER OF THE APPLICATION OF
JOHN J. BRADLEY and BENJAMIN S. FRANKLIN

SWITCH SYSTEM BASE MECHANISM

Board of Appeals
Serial No. 570,331

[Filed December 21, 1978]

DATE	PROCEEDINGS AND ORDERS

Counsel for appellants: Nicholas Prasinos; Ronald T. Reiling Commissioner of Patents & Trademarks

February 12, 1979 PRINTED TRANSCRIPT OF RECORD, filed.

March 8, 1979 BRIEF FOR APPELLANT, filed.

April 9, 1979 Motion of Commissioner of Patents to extend time for filing brief to April 23, 1979, filed.

April 10, 1979 Above motion granted.

April 23, 1979 Motion by Commissioner of Patents and Trademarks to extend time for filing brief to April 27, 1979, filed.

April 24, 1979 Above motion granted

April 27, 1979 BRIEF FOR THE COMMISSIONER OF PATENTS AND TRADEMARKS, FILED.

May 9, 1979 Motion of appellants for extension of time for filing reply brief to May 21, 1979, filed.

May 11, 1979 Above motion granted.

May 11, 1979 REPLY BRIEF FOR APPELLANT FILED.

DATE	PROCEEDINGS AND ORDERS

June 6, 1979 Argued (Judge Cowen, sitting)

July 5, 1979 Reversed, Rich, Judge.

July 27, 1979 FINAL MANDATE ISSUED TO THE COMMISSIONER OF PATENTS AND TRADEMARKS

October 3, 1979 Extension of time granted by the Chief Justice of the Supreme Court in which to file a Petition for Writ of Certiorari to and including December 2, 1979. Solicitor General requested extension of time.

December 3, 1979 Petition for Writ of Certiorari filed in Supreme Court, No. 79-855.

March 17, 1980 Petition for Writ of Certiorari granted. (To be heard same day as 79-527 Diehr et al)

Patent Appeal No. 79-533

IN RE APPLICATION:

JOHN J. BRADLEY and BENJAMIN S. FRANKLIN

Serial No. 570,331

In the matter of the above referenced case which was heard before the United States Court of Custom and Patent Appeals, I was instructed to obtain the tape of the oral arguments from the court, listen to and transcribe the particular portion by the Solicitor.

I hereby certify that the following is a transcription of the portion requested:

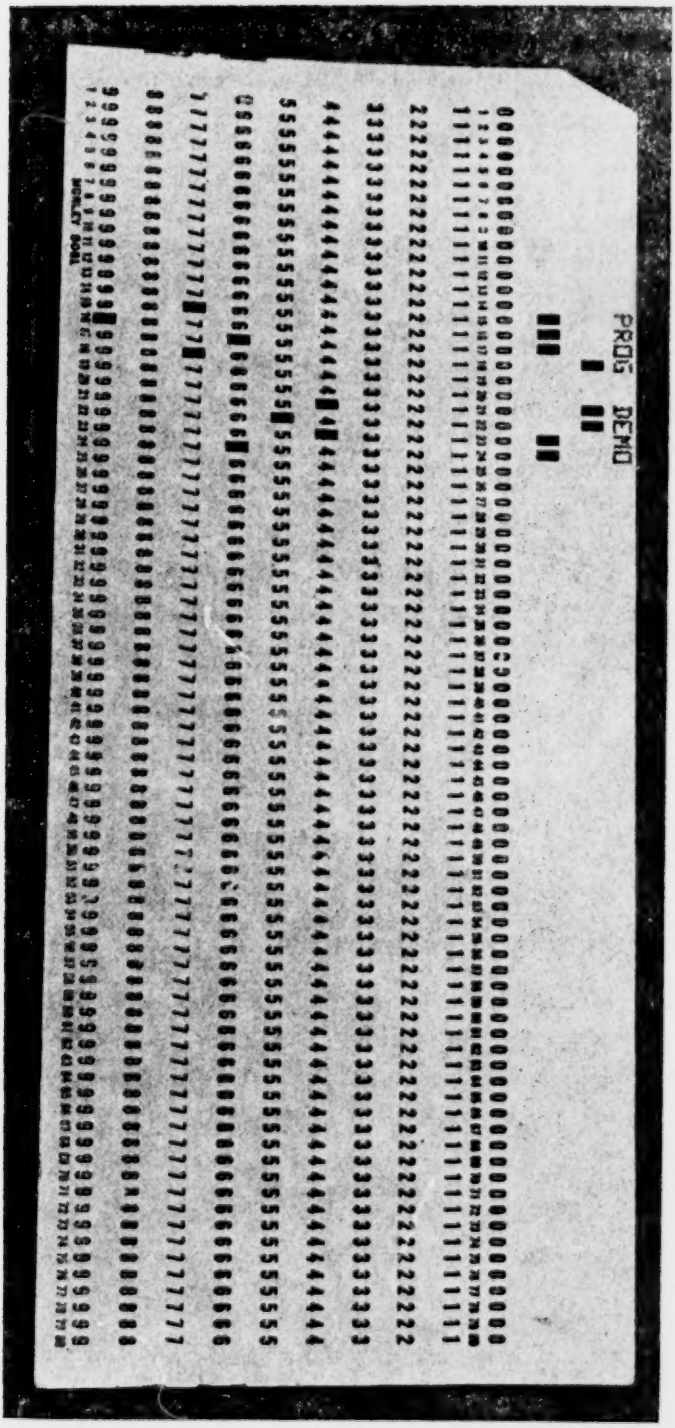
> "Do you agree with Applicant that he is claiming a machine where it starts out, 'In multiprogram computer system . . .'?"

Answer:

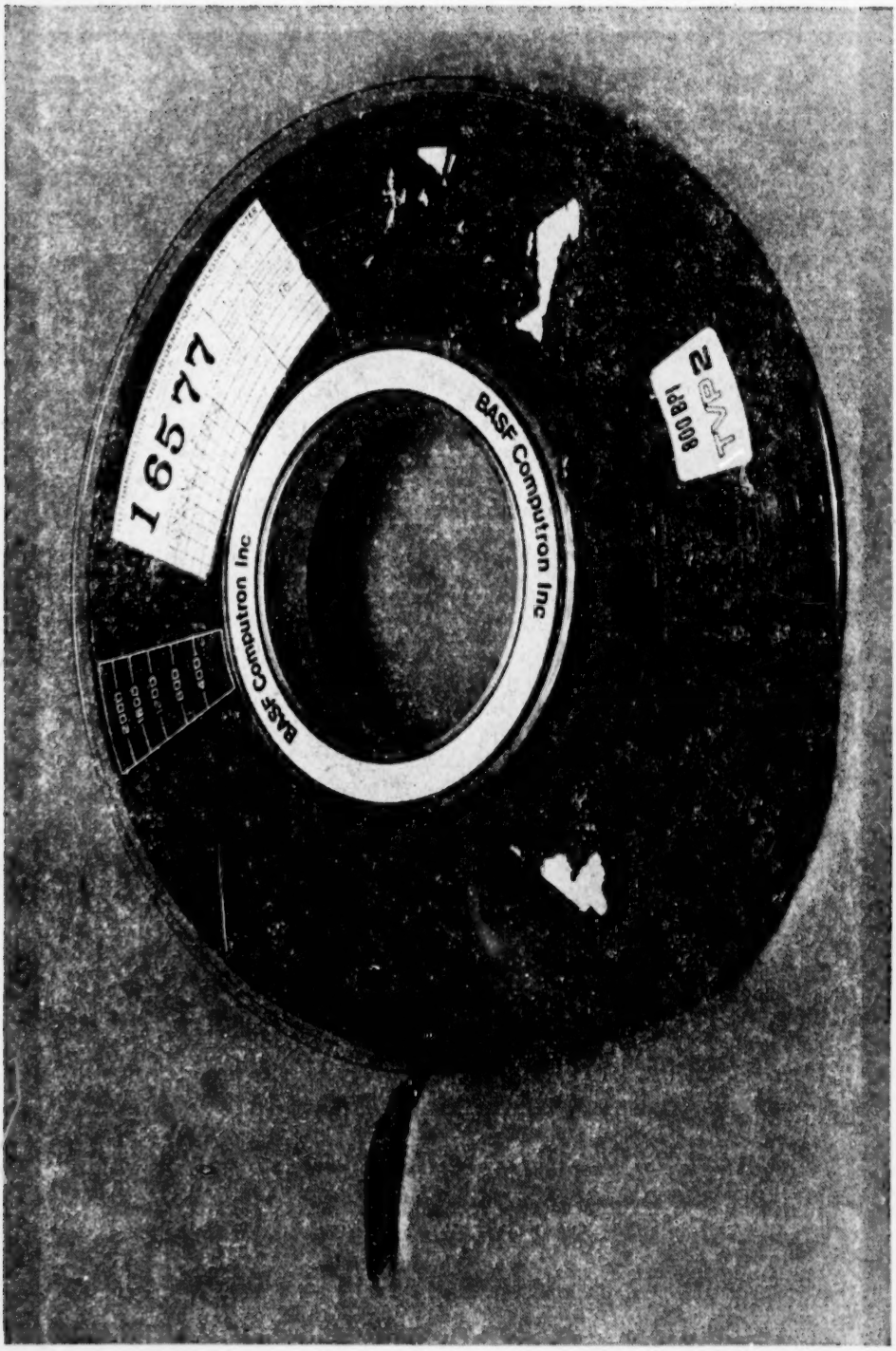
> "I suppose it is a machine, it is an apparatus of some sort, yes Your Honor."

Date: December 7, 1979

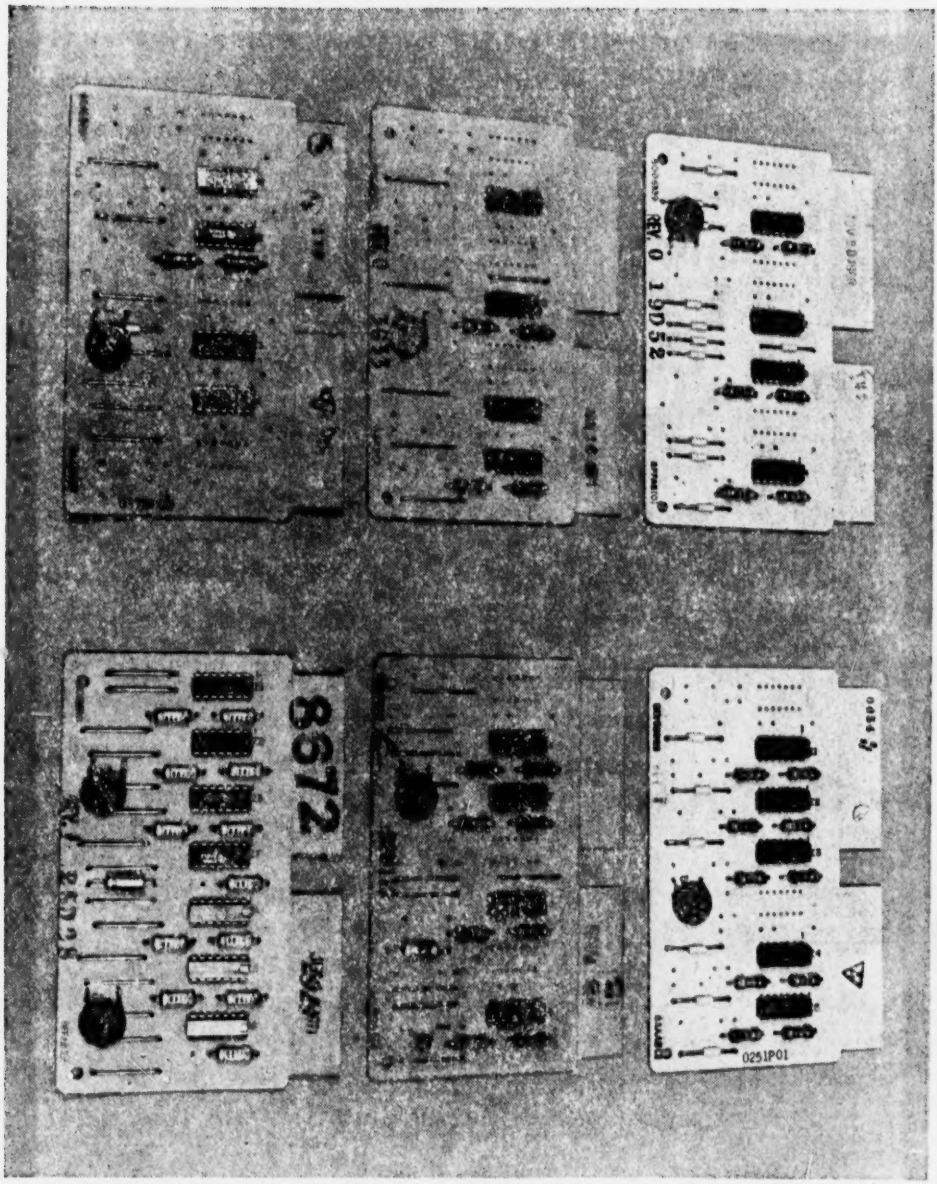
Signed Carolyn L. Leonard
2043 Crystal Plaza Drive
Arlington, Virginia 22202



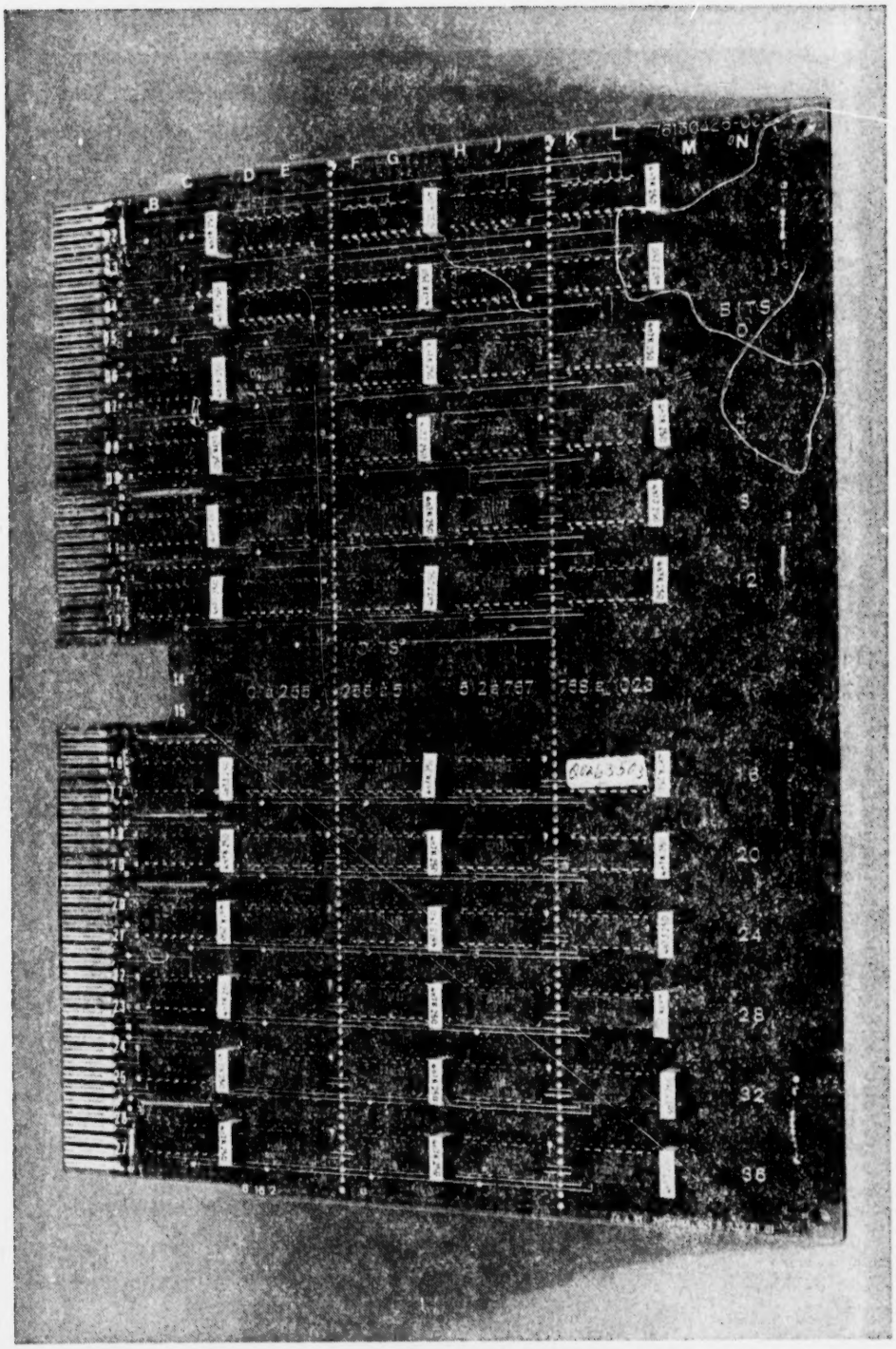
(a) Punch Cards of the type utilized by programmers in computer programming



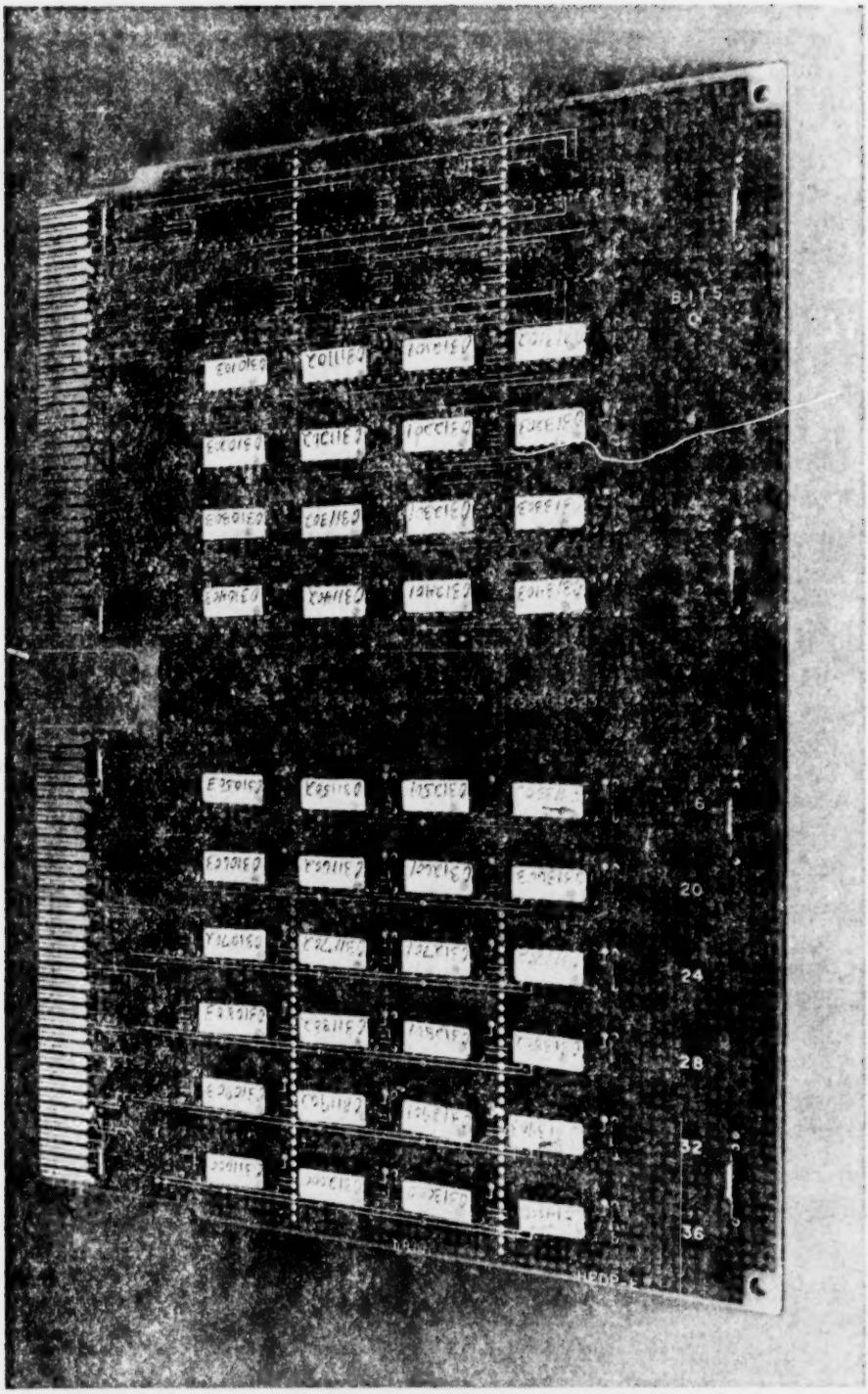
(b) Magnetic Tape of the type utilized by programmers in computer programming



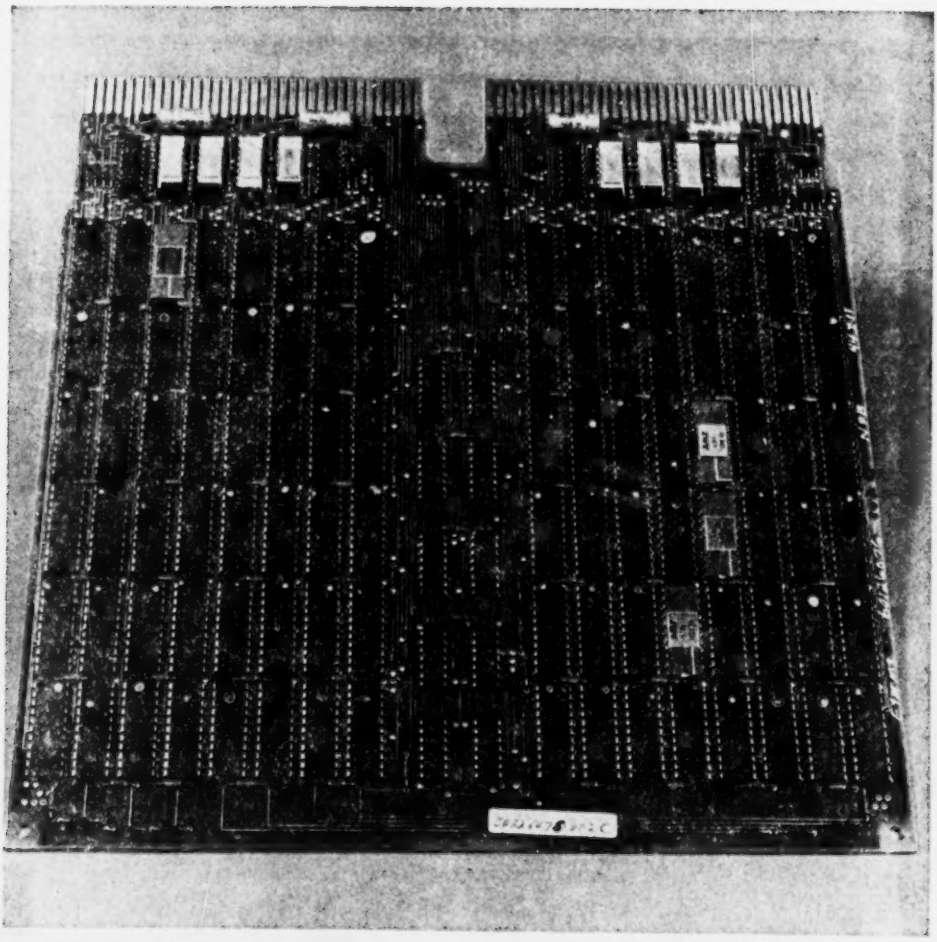
(c) A computer hardwired board of the type utilized by system designers to design new computer machines



(d) Two computer firmware boards of the type utilized by system designers to design new computer machines
Board (1)



(d) Two computer firmware boards of the type utilized by system designers to design new computer machines
Board (2)



(e) A MOS memory board of the type utilized for main memories in compter machines

UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

Appeal No. 79-533

IN THE MATTER OF THE APPLICATION OF
JOHN J. BRADLEY and BENJAMIN S. FRANKLIN

Serial No. 570,331

Decided: July 5, 1979

Before MARKEY, Chief Judge, RICH, BALDWIN, and MILLER, Associate Judges, and COWEN,* Senior Judge.

RICH, Judge.

This appeal is from the decision of the Patent and Trademark Office (PTO) Board of Appeals (board) affirming the rejection of claims 1-6, all of the claims in appellants' application serial No. 570,331, filed April 21, 1975, for "Switch System Base Mechanism," as being drawn to subject matter which is non-statutory under 35 USC 101. We reverse.

The Invention

Appellants' invention is in the field of computer technology. It does not relate to computer applications, i.e., any specific task that a computer is asked to perform, but rather to the internal operation of the computer and its ability to manage efficiently its operation in a multiprogrammed format. A multiprogrammed format is one in which the computer is capable of executing more than one program, and thus perform more than one applica-

* The Honorable Wilson Cowen, U.S. Court of Claims, sitting by designation.

tion at the same time, without the need to reprogram the computer for each task it must perform.

Specifically, the invention relates to altering or repositioning information in the computer's system base. The system base of a computer is a fixed area in main memory which acts as the root for all information structures in the computer. In high performance computer systems, it is very advantageous to store portions of the system base in "scratchpad" registers [1] located in the central processing unit (CPU) of the computer rather than in main memory. This greatly enhances the speed of operation of the computer because the access time (the time it takes to retrieve information from a given place) associated with the scratchpad registers is less than that associated with main memory. A problem arises, however, because a programmer may wish to change the positions or content of information in the system base which happens to be located in the scratchpad registers. These registers are "invisible" to the programmer since, unlike main memory, they cannot be accessed by software (computer programs). Prior art systems altered the system base information resident in the scratchpad registers by either reinitializing the system base (completely reloading the system base containing the new information), a process which consumes a considerable amount of time, or by using software which takes advantage of the model-dependent properties of the particular computer. The latter method has the undesirable effect of resorting to reliance on model-dependent software, which is unacceptable to some computer users.

Appellants' invention enables system base information to be altered without having to resort to these techniques and their accompanying drawbacks. They accomplish their result by employing a "firmware" module, consisting of hardware elements permanently programmed with a

---

[1] A scratchpad register, also known as a scratchpad memory, is a plurality of multibit storage locations, usually located in the central processing unit (CPU) of a computer, used for temporary storage of program information, operands, and calculation results for use by the computer's arithmetic and logic unit, and other information of a temporary nature.

microcode, which directs the data transfers, between the scratchpad registers and the system base located in main memory, which are necessary to effect the alteration.

Claim 1 is representative of the appealed claims:

> 1. In a multiprogramming computer system having a main memory, a central processing unit (CPU) coupled to said main memory, said (CPU) controlling the state of a plurality of groups of processes being in a running, ready, wait or suspended state, said computer system also having scratchpad registers being accessible to an operating system for controlling said multiprogramming computer system, a data structure for storing coded signals for communicating between said processes and said operating system, and said scratchpad registers, said data structure comprising:
>
> (a) first means in said data structure and communicating with said operating system for storing coded signals indicative of an address for a selected one of said processes;
>
> (b) second means in said first means for storing coded signals indicating priority of said selected one of said processes in relation to others of said processes for obtaining control of said CPU when ready;
>
> (c) third means in said data structure and communicating with said operating system, for storing coded signals indicative of an address for a selected one of said plurality of groups of processes, and,
>
> (d) fourth means coupled to said data structure and said scratchpad registers, for generating signals causing the changing of information in said data structure and said scratchpad registers.

The Rejection

The examiner rejected the appealed claims on the authority of *Gottschalk* v. *Benson*, 409 U.S. 63, 175 USPQ 673 (1972) (hereinafter *Benson*) before the Supreme

Court's decision in *Parker* v. *Flook*, 437 U.S. 584, 198 USPQ 193 (1978) (hereinafter *Flook*). In his final rejection, dated October 27, 1976, the examiner stated that the subject matter "deemed as the invention" is "a data structure" and then made the following analysis:

> The term "data structure" as recited in the claim is comprised of four entities described as "first means", "second means", "third means", and "fourth means." Whether these so called "means" are hardware structure which store coded signals for performing a particular function or whether they refer to coded signals representing computer control words per se, any life which the claim may have in view of that which is admittedly old in the art, resides in the "technique" (page 5, line 4) or "switch-system base instruction" (page 5, lines 13 and 14) for solving the problem of changing information of the system base which is located in scratch-pad registers which software cannot access (page 4, line 20). The invention then is embodied in the coded signals representing instructions to the computer, the running process word RPW (first means of claim), priority level indicator word PRI (second means), J Table word and G Table word (third means), and micro-instructions of Figures 15c [sic, 15b] and 15c as found in the central store 1301. The invention resides in a "data structure" or an algorithm designed to control the multiprogramming computer to solve the particular problem indicated.
>
> Under the ruling in *Gottschalk* v. *Benson*, 175 USPQ 673, the instant claims, depending upon a program implemented algorithm for patentability, are deemed nonstatutory subject matter.

Appellants requested reconsideration and argued that their claims are directed to data structures in hardware which are "specific new, novel and unobvious [in] arrangement." They asserted that by stating that the invention resided in a "technique," the examiner was clearly disregarding the claims and interpreting the invention strictly on the basis of what is found in the specification,

because no "technique" is *claimed*. They stated that even if a technique (i.e., process) were claimed, *Benson* does not render all such inventions nonstatutory, and that their invention does not involve a *mathematical* algorithm.[2]

In his Answer before the board, the examiner noted that all of the limitations found in claim 1 were old in the art and that the "claim is thus reciting prior art coupled with subject matter which the U.S. Supreme Court has found to be non-statutory in *Benson*." Appellants responded in their reply brief that it makes no difference whether individual elements are old in the art and that it is the elements in combination which define the invention as a whole.

The Board

The board rendered its decision on September 20, 1978, after the Supreme Court's decision in *Flook*. After incorrectly stating that none of the claims recited the term "firmware," [3] the board analyzed appellants' claims element by element, concluding that the only novel ar-

---

[2] In *Benson*, the Supreme Court was clearly limiting its discussion to mathematical algorithms:

> The patent sought is on a method of programming a general purpose digital computer to convert signals from binary coded decimal form into pure binary form. *A procedure for solving a given type of mathematical problem is known as an "algorithm." The procedures set forth in the present claims are of that kind; that is to say, they are a generalized formulation for programs to solve mathematical problems of converting one form of numerical representation to another.* [409 U.S. at 65, 175 USPQ at 674 (emphasis ours).]

It is not clear whether the examiner regarded appellants' invention as a mathematical algorithm, but, as will become evident, the board did so regard the invention.

[3] Claim 3, as well as dependent claims 4-6, explicitly recites firmware. Firmware is a term of art in the computer field and refers to microinstructions permanently embodied in hardware elements. For a further discussion, *see generally* Ross, *The Patentability of Computer "Firmware,"* 59 JPOS 731 (1977). We need not and do not decide at this time whether firmware *per se* is statutory under 35 USC 101 because the invention *as a whole* is not directed thereto. See our further discussion, infra.

rangement of the recited structures resided in the microprogramming, "which together with its attendant memory hardware appears to constitute firmware."

Apparently on the basis of *Flook*, the board affirmed the rejection because it was of the opinion that the appealed claims are directed to a method of calculation or mathematical algorithm. The board found the claims similar to the claims at issue in *In re Waldbaum*, 559 F.2d 611, 194 USPQ 465 (CCPA 1977), which included language characterized by the board as "obviously related to calculating and mathematical problem solving."[4] Although the claims here at issue do not contain similar mathematical language, the board said that this "does not make the functions attendant the 'means' of appellants' claims any less mathematical or less related to an algorithm within the meaning assigned that term by the USSC in *Benson*."

To support its conclusion that the appealed claims are mathematical in nature, the board relied on a statement in the specification to the effect that all of the data in the computer are in binary form, but may be interpreted as binary coded decimal, decimal or alphanumeric. We reproduce the board's reasoning in full:

> Since digital computers normally operate in some numerical radix, binary, binary coded decimal, or the like, we consider the operation of appellants' claimed invention to be mathematical.
>
> Every operation performed in appellants' invention as claimed involves the accommodation of data and instructions to the size of the registers in memory, and to the positional assignment to the registers in memory by the use of some numerical measure or quantity effected by way of electrical signals. In

---

[4] Waldbaum claim 1 reads in pertinent part:

> a method * * * to *count the number* of busy lines * * * *comparing means* * * * *to derive the number of l's* in said data word * * *. [Emphasis ours.]

It is clear that this claim recites a mathematical algorithm. It solves a mathematical problem, to wit, *counting* a number of busy lines in a telephone system.

> whatever form the instructions employed in appellants' invention are characterized, numerical or otherwise, we think it is accurate to say that the operation of appellants' structure is mathematical and that the instructions constitute a procedure which is algorithmic in character, to the same degree as that of the Waldbaum structure and that satisfactory operation of the apparatus claimed represents the successful solution of a mathematical problem falling within the definition of algorithm supplied in *Benson* and reiterated in *Flook*.
>
> Here the method of calculation is to be used in the internal operation of the computer system and the claims recite such an end use by means of the functions recited for the various means. A claim for an improved method of calculation, even when tied to a specific end use, is unpatentable subject matter under 35 USC 101, *Flook*, *supra*.

In summary, the board stated that the claims are drawn to apparatus in form only, and couple the apparatus (which it asserts is old in the art) "with subject matter, namely, programming, which is nonstatutory under the *Benson*, *Christensen*,[5] *Waldbaum* and *Flook* cases * * *."

OPINION

The examiner's basis for the rejection is grounded on the erroneous interpretation of the Supreme Court's decision and opinion in *Benson*, namely, that all computer program or program-related inventions are nonstatutory under § 101. Both the Supreme Court[6] and this

---

[5] *In re Christensen*, 478 F.2d 1392, 178 USPQ 35 (CCPA 1973).

[6] In *Benson*, the Court stated, 409 U.S. at 71, 175 USPQ at 676 (emphasis ours):

> It is said that this decision precludes a patent for any program servicing a computer. *We do not so hold.*

In *Flook*, the Court stated, 437 U.S. at 595 n.18, 198 USPQ at 199 n.18:

> Very simply, our holding today is that a claim for an improved method of calculation, even when tied to a specific end use, is unpatentable subject matter under § 101.

court[7] have thoroughly repudiated this view. Our decision, therefore, is based solely on the analysis made by the board.

The board said that claims do not *directly* recite a mathematical formula, algorithm, or method of calculation, but, nevertheless, held the claims to be mathematical in nature. As appears from the quoted portion of the board opinion, the board regarded the fact that digital computers operate in some number radix as conclusive on the issue of whether the appealed claims recite a mathematical algorithm in the *Benson* and *Flook* sense. The board did not, however, direct attention to any specific formula it thought is utilized, or to what, if anything, the mathematical calculations alleged to be present in the claims are directed.

We do not agree with the board. We are constrained to reject its reasoning. Such reasoning leads to the conclusion that any computer-related invention must be regarded as mathematical in nature, a conclusion which is not compelled by either *Benson* or *Flook*.

The board's analysis confuses *what* the computer does with *how* it is done. It is of course true that a modern digital computer manipulates data, usually in binary form, by performing mathematical operations, such as addition, substraction, multiplication, division, or bit shifting, on the data. But this is only *how* the computer does what it does. Of importance is the significance of the data and their manipulation in the real world, i.e., *what* the computer is doing. It may represent the solution of the Pythagorean theorem, or a complex vector equation describing the behavior of a rocket in flight, in which case the computer is performing a mathematical algorithm and solving an equation. This is what was in-

---

[7] *See In re Gelnovatch*, 595 F.2d 32, 36-37, 201 USPQ 136, 141 (CCPA 1979); *In re Johnson*, 589 F.2d 1070, 1075, 200 USPQ 199, 205 (CCPA 1978); *In re Sarkar*, 588 F.2d 1330, 1333, 200 USPQ 132, 137 (CCPA 1978); *In re Freeman*, 573 F.2d 1237, 1244, 197 USPQ 464, 470 (CCPA 1978); *In re de Castelet*, 562 F.2d 1236, 1240, 195 USPQ 439, 443 (CCPA 1977); *In re Chatfield*, 545 F.2d 152, 155-56, 191 USPQ 730, 733-34 (CCPA 1976), *cert. denied*, 434 U.S. 875, 195 USPQ 465 (1977).

volved in *Benson* and *Flook*. On the other hand, it may be that the data and the manipulations performed thereon by the computer, when viewed on the human level, represent the contents of a page of the Milwaukee telephone directory, or the text of a court opinion retrieved by a computerized law service. Such information is utterly devoid of mathematical significance. Thus, the board's analysis does nothing but provide a quick and automatic negative answer to the § 101 question simply because a computer program is involved.

Appellants have continuously insisted that they are claiming a new and unobvious combination of *hardware elements*, i.e., a new machine or apparatus.[8] The issues of novelty and unobviousness[9] are not before us, but we agree with appellants insofar as they characterize their invention as a machine or apparatus. The board likewise seems to agree on this point. In its opinion, it identifies all of the "means" of appellants' claim 1 as structural hardware elements, such as registers, portions of main memory and control store,[10] and other computer components. Thus, the claim falls literally within the boundaries of § 101.

Appellants have characterized their combination of hardware elements as a mechanism which enables the computer to alter information in its system base in a manner not previously possible. They are in no way claiming the altered information; in fact, the particular information acted upon by appellants' invention is irrelevant to the operation of the invention itself. We see no difference in this regard, with respect to being within

---

[8] 35 USC 101 states:

> Whoever invents or discovers any new and useful process, *machine*, manufacture, or composition of matter, *or any new and useful improvement thereof*, may obtain a patent therefor, subject to the conditions and requirements of this title. [Emphasis ours.]

[9] 35 USC 102, 103.

[10] The control store of a computer is a plurality of multibit storage locations (memory) containing microinstructions, which, when decoded by appropriate circuitry, provide control signals which cause specific operations to take place in the computer's CPU.

§ 101, between appellants' claimed invention and a strictly mechanical adding machine, which is certainly statutory if claimed in a manner which does not embrace any particular calculation that the machine is capable of making.

The PTO regards as significant the fact that firmware is involved in the present invention. In a sense, firmware may be likened to software (computer programs); it is information which has been embodied into hardware by, for example, destroying fusible links in a read only memory (ROM) array. In appellants' invention, the information contained within the firmware, which is located in the control store of the computer, directs the desired information transfers within the system base.

If appellants were claiming the information embodied in the firmware or the firmware itself, per se, a different case would be presented. We express no opinion on the statutory nature of such an invention, a question not before us. Appellants are claiming a combination of hardware elements, one of which happens to be a portion of the computer's control store microprogrammed in a particular manner. It is this subject matter with which we must deal.

From our reading of appellants' specification, their claimed "data structure" is merely that which results from the arrangement of the recited hardware elements in the claimed manner. It is the result of certain structural "means" performing certain recited functions as explicitly sanctioned by 35 USC 112, sixth paragraph. We disapprove the board's distillation of appellants' claim down to the information contained in the firmware in order to hold it nonstatutory. The invention is not claimed in that manner, and, in this case, we see no reason to view the claim format as a subterfuge for masking the presence of an essentially nonstatutory invention.

Even though the claimed invention is a machine, we must nevertheless determine whether the claim recites a mathematical algorithm, and, if so, whether it preempts the use of the algorithm. *In re Noll*, 545 F.2d 141, 148, 191 USPQ 721, 726 (CCPA 1976). In doing so, we apply the two-step test in *In re Freeman*, supra note 7.

When we examine appellants' invention as a whole under the first step of this test, including the information microprogrammed into the firmware element as depicted in Figs. 14(a-i) and 15(b-c), we fail to detect the presence of any mathematical algorithm. In altering information in the system base as desired, certain "calculations" are made, such as determining whether a given quantity is equal to zero, or, as noted by the solicitor, multiplying an address in memory by sixteen to arrive at another address. However, it certainly cannot be said that comparing with zero or multiplying by sixteen is preempted by appellants' claims. Furthermore, the presence of these calculations does not transform the invention as a whole into a method of calculation. *Cf. In re Gelnovatch*, supra note 7. There is no solution of an equation, such as the new alarm limit in *Flook*, or the equivalent pure binary number, as in *Benson*, present in the computer after the task has been completed. *See In re Chatfield*, supra note 7.

In summary, we have examined the claims thoroughly and we do not find any mathematical formula or mathematical method of calculation, improved or otherwise, which is either claimed as such or essential to the vitality of the claims. We find that the invention is a combination of tangible hardware elements—a machine—including some hardware elements which contain microprogrammed information termed "firmware." We do not find the invention to be nonstatutory under the authority of *Benson* or *Flook*, or under the authority of our own cases, such as those cited at note 7 ante. Therefore, the decision of the board is *reversed.*

REVERSED

UNITED STATES COURT OF CUSTOMS
AND PATENT APPEALS

MANDATE

No. 79-533

IN THE MATTER OF THE APPLICATION

OF

JOHN J. BRADLEY and BENJAMIN S. FRANKLIN

Serial No. 570,331

ON APPEAL from the Board of Appeals

This CAUSE having been heard and considered, it is

ORDERED and ADJUDGED: Reversed

Dated July 5, 1979

A True Copy
GEORGE E. HUTCHINSON,
Test:
Clerk, United States Court of
Customs and Patent Appeals

Certified this 27th day of July 1979

/s/ George E. Hutchinson
GEORGE E. HUTCHINSON

IN THE UNITED STATES COURT OF
CUSTOMS AND PATENT APPEALS

Patent Appeal Docket No. 79-533

IN RE APPLICATION OF
JOHN J. BRADLEY and BENJAMIN S. FRANKLIN

APPEAL FROM BOARD OF APPEALS

Serial Number 570,331

PRAECIPE FOR TRANSCRIPT OF RECORD, NOVEMBER 17, 1978

November 13, 1978

Commissioner of Patents & Trademarks
Washington
District of Columbia 20231

Dear Sir:

Request is hereby made that a transcript of the record be prepared and filed in the United States Court of Customs and Patent Appeals for use in connection with the appeal to the United States Court of Customs and Patent Appeals from the decision of the Patent Office Board of Appeals rejecting the claims of the above-identified patent application, the said transcript to include the following:

(1) This praecipe for transcript of record.
(Serial Number 570,331, filed April 21, 1975)

(2) Application Serial No. 570,331 (including petition specification, drawings and declaration or oath as filed, but not including the claims).

(3) Letter of the Examiner (First Office Action) March 12, 1976 (5 pages).

(4) Request for Extension of Time, May 18, 1976, and grant thereof.

(5) Amendment A, August 10, 1976.

(6) Letter of the Examiner (Second Office Action) October 27, 1976.

(7) Amendment Under Rule 116, December 28, 1976.

(8) Advisory Action, February 1, 1977.

(9) Notice of Appeal to Board of Appeals dated October 27, 1976.

(10) Brief for Appellant.

(11) Examiner's Answer Mailed August 4, 1977.

(12) Appellants' Reply Brief to Examiner's Answer.

(13) Notice of Hearing, Mailed December 13, 1977.

(14) Oral Hearing Confirmation dated January 16, 1978.

(15) Transcript of Oral Hearing (If Available).

(16) The claims on appeal (a copy of such claims are on pages 6-8 of the Brief for Appellant).

(17) Decision of Board of Appeals mailed September 20, 1978.

(18) Notice of Appeal to the Court of Customs and Patent Appeals and Reason for Appeal dated

It is requested that the Patent Office telephone the undersigned when the certified transcript of the record is ready so that the record can be picked up and delivered to the Clerk of the Court of Customs and Patent Appeals.

It is further requested that costs for preparation of the transcript be charged to Honeywell Information Systems Inc. Deposit Order Account No. 08-2732.

Respectfully submitted,

By: /s/ Nicholas Prasinos
NICHOLAS PRASINOS
Attorney for Applicants
Registration No. 24,791

APPLICATION OF JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN, FILED APRIL 21, 1975, SERIAL NUMBER 570,331, FOR SWITCH SYSTEM BASE MECHANISM.

ABSTRACT OF THE DISCLOSURE

A model-independent instruction in firmware is provided to change, upon execution, information in the system base which is located in both main memory and internal scratch pad registers.

RELATED APPLICATIONS

The following applications are incorporated by reference to the instant application.

1. "Buffer Store" invented by J. L. Curley, T. J. Donahue, W. A. Martland, and B. S. Franklin, filed on October 5, 1972 having Serial Number 295,301 and assigned to the same assignee named herein.

2. "Variable Masking for Segmented Memory" invented by Wallace A. Martland and John L. Curley, filed on October 5, 1972 having Serial Number 295,303 and assigned to the same assignee named herein.

3. "Override Hardware for Main Store Sequencer" invented by Thomas J. Donahue, filed on October 5, 1972 having Serial Number 295,418 and assigned to the same assignee named herein.

4. "Main Memory Sequencer" invented by T. J. Donahue, J. L. Curley, B. S. Franklin, W. A. Martland, and L. V. Cornaro, filed on October 5, 1972 having Serial Number 295,331 and assigned to the same assignee named herein.

5. "Main Memory Reconfiguration" invented by J. L. Curley, B. S. Franklin, W. A. Martland, T. J. Donahue and L. V. Cornaro filed on October 5, 1972 having Serial Number 295,417 and assigned to the same assignee named herein.

6. "Ring Checking Hardware" invented by ———— filed on the same date as the instant invention, having Serial Number ———— and assigned to the same assignee named herein.

7. "Segmented Address Development" invented by ——————, filed on ———— and having Serial Number ———— and assigned to the same assignee named herein.

8. "Protection of Information in a Multiprogram Multiprocessor Computer System" invented by ————, filed on ———— and having Serial No. ————, and assigned to the same assignee named herein.

9. "Ring Crossing Mechanism" invented by ————, filed on ————, and having Serial Number ———— and assigned to the same assignee named herein.

10. "Procedure Calls and Stack Operations", invented by ————, filed on ————, and having Serial Number ———— and assigned to the same assignee named herein.

11. "Process Synchronization Utilizing Semaphores" invented by ————, filed on ————, and having Serial Number ————.

12. "Arrangement for Automatically Managing, Scheduling and Allocating System Resources" invented by ——————, filed on ———— and having Serial Number ———— and assigned to the same assignee named herein.

13. "Data Processing Memory Management System" invented by ————, filed on ———— and having Serial Number ———— and assigned to the same assignee named herein.

14. "Arrangement for Initializing a Computer System Controlled by Reloadable Site-Dependent Information Structures", invented by ————, filed on ———— and having Serial Number ———— and assigned to the same assignee named herein.

15. "System Base", invented by ————, filed on ————, and having Serial Number ———— and assigned to the same assignee named herein.

16. "Absolutization During Initialization", invented by ————, filed on ————, and having Serial Number ———— and assigned to the same assignee named herein.

BACKGROUND

Field of the Invention:

This invention relates generally to computer systems and more particularly to a system and method for communicating between hardware and software in a multiprogramming/multiprocessing environment.

Multiprogramming/multiprocessing computer systems require the use of a communications area in main storage for communications between hardware and software. (See the application entitled "System Base" invented by ———— filed on ———— having Serial Number ———— and assigned to the instant assignee, and which application has been incorporated by reference herein). The system base is generally located at a fixed location in main memory, although this location may be changed or relocated through the use of a device such as a boundary address register (BAR). (See invented by ————, filed on ————, having Serial Number ———— and assigned to the instant assignee, and further incorporated by reference in the instant application).

High performance computing systems find it necessary to store portions of the system base in internal scratchpad registers instead of main memory, and hence inaccessible and invisible to software running in the system. Accordingly, a problem results when software (i.e. the programmer) wants to change positions or change information of the system base which happens to be located in these internal central processor scratchpad registers which software cannot access.

Prior Art:

The problem has been solved in one of two ways. The first method is to simply re-initialize the entire system, loading the system base with different contents as a result of a new initialization process. This technique is extremely time-consuming and quite undesirable.

A second technique for solving the problem is to have the software employ special model-dependent properties

of the particular machine on which it was running, to change the system base. This results in the very undesirable property of completely model-dependent software—a feature which certainly does not serve to endear a computer manufacturer to a customer.

What is needed therefore is a technique of changing information in the system base both in main memory and internal scratchpad registers making the system base in scratchpad memory accessible to software but not visible to software i.e. cannot be read by the software.

OBJECTS OF THE INVENTION

It is a primary object of the invention to provide an improved system and method for changing information in the system base of a computer system.

It is another object of the invention to provide a switch-system base instruction in firmware which changes, upon execution, information in the system base.

It is still another object of the invention to provide a system and method for changing information in the system base which is located in main memory and scratchpad registers.

Yet another object of the invention is to provide an improved switch system base mechanism.

These and other objects of the invention will become apparent from the description of a preferred embodiment of the invention when read in conjunction with the drawings contained herewith.

SUMMARY OF THE INVENTION

A model-independent instruction in firmware is provided to change, upon its execution, information in the system base which is located in both main memory and internal scratchpad registers.

The system base located at a location specified by the boundary address register BAR in main memory, is copied into a new location in main memory with predetermined portions of its information changed. This is readily accomplished by the operating system. (See ap-

plication incorporated by reference into the application herein). The switch system base instruction then transfers the information which is in the system base at the new location into the system base at the location specified by BAR and also that portion of the system base located in scratchpad memory.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features which are characteristic of the invention are set forth with particularity in the appended claims. The invention itself, however, both as to organization and operation together with further objects and advantages thereof may best be understood by references to the following description taken in conjunction with the drawings in which:

Figure 1 is a block diagram of a multiprogramming system utilizing the invention.

Figure 2 is a schematic representation of various hardware structures utilized by the invention.

Figure 3 is a legend of terms used for reserved areas of storage in registers depicted in Figure 2.

Figure 4 is a schematic diagram of a process control block.

Figure 5 is a schematic diagram of a system for addressing a process control block.

Figure 6 is a schematic diagram of the system base of the invention.

Figure 7A and 7B are a schematic representation of a stack segment and a stack frame respectively.

Figure 8 is a schematic diagram of a system for addressing G-segments and in particular the queue of processes in the GO segment.

Figure 9 is an exploded schematic diagram of a GO segment illustrating queue of processes and process linking.

Figures 10a through 10L are block diagrams of structures in the PCB.

Figures 11a through 11r are block diagrams of structures in the system base.

Figure 12 is a schematic diagram of addressing schemes of user and system segments utilizing the system base and PCB structures.

Figure 13 is a schematic diagram of the control unit.

Figures 14a through 14i are flow diagrams of the dispatcher unit in firmware.

Figure 15a is a structure of the switch system base instruction.

Figures 15b-15c are flow diagrams of the switch system base instruction in firmware.

DESCRIPTION OF A PREFERRED EMBODIMENT

GENERAL DISCUSSION

The invention operates typically in the hardware system environment hereinafter described coordinated by a hardware/firmware/software operating system. Referring to Figure 1 the subsystems are the processor subsystem 101, the storage subsystem 102, and one or more —up to 32—peripheral subsystems 103. The processor subsystem contains a central processing unit (CPU) 104 and up to four input/output control units (IOC) 105. Each peripheral subsystem consists of a peripheral control unit (PCU) 106, a number of device adapters (DA) 107, and up to 256 peripheral i/o devices 108. The storage subsystem 102 consists of one to four semiconductor memory modules of 32 to 512 kilobytes each.

I. PROCESSOR SUBSYSTEM

In the processor subsystem 101, the CPU 104 performs the basic processing operations for the system, and interfaces with memory 102. The IOC 105 controls all information exchanges between the storage subsystem 102 and peripheral devices 106.

A. CENTRAL PROCESSING UNIT

The CPU includes a main memory synchronizer 109, a buffer store 110, various elements that comprise the

computational unit 111, and optional emulation facilities 112. The main memory synchronizer 109 resolves conflicts for the use of main memory among the computational unit 111, the buffer store 110, and the IOC 109. Conflicts are resolved on a priority basis: the IOC has the highest priority followed by memory writes (from the computational unit) and then memory reads (into the buffer store). The main CPU also includes the address control unit ACU 131 which controls main memory addressing and the associative memory AS 132 used to store most recently used addresses of main memory. The buffer store 110 is a small high-speed buffer memory that reproduces a selected region of main memory and interfaces with the computational unit to decrease average memory access time. During each memory read, both the buffer store and main memory are accessed. If the information to be fetched is already in the buffer store, the main memory read is terminated and the information fetched from the buffer store. Otherwise the main memory 102 is read. Every time this is done, the CPU 104 fetches 32 bytes that contains the desired information. This information remains in the buffer store for future memory references. Since the buffer store is transparent to software, the program controlling the computer at any given moment cannot determine whether the information it is processing has been fetched from the buffer store or from main memory.

The computational unit 111 performs all data processing and address generation within the CPU. A typical control store 130 within the computational unit (see a book entitled Microprogramming: Principles and Practices, Samir S. Husson, Prentice Hall, Inc.) contains firmware which initializes the system, controls the CPU 104 and IOC 105, and decodes an instruction set (not shown). Optionally the control store may provide scientific instructions, test routines, emulation packages, or special purpose features which extend the capabilities of the processor subsystem.

As an option, the CPU provides emulation of systems other than the instant system. Emulators 112 are components of firmware, software, and in some instances hardware.

B. INPUT-OUTPUT CONTROL UNIT

The IOC 105 portion of the processor subsystem provides a data path between any peripheral subsystem 103 and the storage subsystem 102. This path allows for the initiation of peripheral commands and controls the resulting data transfers. An IOC can typically handle up to 32 channel control units (not shown).

C. PERIPHERAL SUBSYSTEMS

In a peripheral subsystem 103 on Figure 1 the PCU 106 is a stand-alone microprogramming processor that relieves the load on the CPU 104 by controlling the i/o devices 108 during i/o operations. The PCU does this by executing instructions contained in a channel program. This program results in arithmetic, logical, transfer, shift, and branch operations being performed in the PCU. There are several kinds of PCU's according to the kind of device each controls: i.e. unit record, mass (disk) storage, magnetic tape, communications, etc.

Device adapters 107 mediate between every PCU and the devices it controls. Each contains the dedicated firmware and logic necessary to implement communication with a particular type of device. Depending on the type, a DA 107 controls one or several devices.

The major functions performed by a peripheral subsystem 103 are as follows:

1. Transforming CPU instructions into a series of commands acceptable to the appropriate peripheral device.
2. Packing and unpacking data in the form needed by the CPU or the appropriate peripheral device.
3. Keeping the CPU informed of the status of the subsystem and of the devices under its control.
4. Independently initiating and processing error and recovery procedures.
5. Allowing on-line diagnosis of a device without disturbing the device-sharing capabilities of the associated peripheral processor.

A PCU resolves conflicts for main memory between devices attached to it; however, the IOC resolves conflicts between PCU's.

D. STORAGE SUBSYSTEM

Each memory module 1-4 is 4 or 8 bytes wide. The number of modules, their size, and the data path width may vary according to size of computer. Memory modules are four-way interleaved in such a way that the four modules are accessed sequentially (module 1 contains the first 8 bytes, module 2 contains the second 8 bytes, etc.). Interleaving decreases the number of conflicts for access to main memory and thereby decreases the average memory access time. Memory is reconfigurable in case of failure; i.e., blocks of memory within a module may be removed without destroying contiguous addressing.

Main memory 102 consists of a capacitive storage medium in the form of metal oxide semiconductor (MOS) chips. This medium operates on the refresh principle to maintain information. Each memory location is typically refreshed at least once every 2 milliseconds; the design ensures that few conflicts occur between refresh timing and memory accesses. (In cases of conflict, refreshing takes precedence).

An area at the beginning of main memory is reserved for hardware and firmware. The upper limit of this area is defined by the content of a boundary address register (BAR—to be later described) which is visible to the system software. The BAR content is set at system initialization time. The memory area below the address specified in the BAR can contain IOC tables which define the configuration of the peripheral subsystems, firmware to control the CPU, or microprograms and tables for emulation. The size of the area below the address specified in the BAR depends on the system configuration. Whether microprograms are in main memory or control store depends on the system configuration and the applications run on the system.

II. BASIC MACHINE STRUCTURES

There are typically three basic data structures utilized in this hardware: data formats, software visible registers, and the instruction formats.

A. DATA FORMATS

Information is transferred between memory and the CPU in multiples of 8 parallel bits. Each 8-bit unit of information is called a byte. Parity or error correction data is also transferred with data but cannot be affected by software. Therefore, in this patent specification the term data excludes the associated parity or error correction data.

B. BYTES

Bits within a byte are numbered 0 through 7 from left to right. Bytes are processed separately or in groups. Two bytes constitute a halfword, 4 bytes a word, 8 bytes a doubleword, and 16 bytes a quadword. These are the basic formats for all data, including instructions.

C. DATA REPRESENTATION

All data are in binary form, but may be interpreted as binary, decimal, or alphanumeric. Data bits are interpreted in groups of four, as binary coded decimal data; eight as alphanumeric, or 16 to 64 as binary digits. The latter are interpreted as signed, fixed, or floating-piont numbers in binary notation. Any number of contiguous bits up to a doubleword may also be manipulated as a string. The alphanumeric character set is represented in EBCDIC. ASCII is supported as an alternate exchange code.

D. BYTE ADDRESSES

Byte locations in main memory are consecutively numbered starting with zero; each number is the address of the byte. A group of consecutive bytes is said to be

halfword-, word-, doubleword-, or quadword-aligned, if the address of the left byte in a group is a multiple of 2, 4, 8, or 16, respectively. Whenever a halfword, word, doubleword, or quadword is so aligned, that unit can be fetched from that address. The location of data in main memory is specified by a data descriptor which is accessed indirectly during address development. (See Patent Application No. ———— filed ———— entitled Segmented Address Development and assigned to the same assignee as the instant application).

E. VISIBLE REGISTERS

There are 33 user-visible registers in the CPU 104 Figure 1 whose contents collectively define the state of the CPU. There are four types: (See Figure 2).

1. general registers
2. base registers
3. scientific registers (optional)
4. miscellaneous registers

F. GENERAL REGISTERS

General registers (GR) 201 are used to manipulate fixed-point binary numbers and bit strings. There are typically sixteen 32-bit general registers in the CPU 104—GR0 through GR15. General register GR8 through GR15 are also usable as index registers. When used as index registers, they are herein called X0 through X7: Indexing is performed using the 32-bit two's component integer contained in a register.

G. BASE REGISTERS

Base registers (BR) have the same format as instruction counters IC and stack registers 202-203. Base registers are used during address computation to define a part of memory. There are typically eight 32-bit base registers, BR0 through BR7.

H. SCIENTIFIC REGISTERS

Scientific registers (SR) are optional equipment for computation with floating-point binary numbers. There are typically four 8-byte scientific registers which are referred to as SR0 through SR3. Scientific registers have the format 204-205 of Figure 2.

I. MISCELLANEOUS REGISTERS

There are five other registers:

- instruction counter—having format 202-203;
- status register—having format 207;
- stack register (called the T register);
- boundary address register—having format 202-203; and
- hardware control mask register—having format 208.

The instruction counter (IC) is a 32-bit register that contains the address of the instruction being executed. The status register (STR) 207 is an 8-bit register that records facts about the procedure currently being executed, for example, whether an underflow was caused by the most recent operation. The stack register also known as the T-register is a 32-bit register that contains a pointer to the top of a pushdown stack associated with the currently active procedure. Stacks to be described infra provide a work space, and a mechanism for saving local variables and preserving procedure entry, and return information. The boundary address register (BAR) 206 is a 28-bit register which specifies the lowest absolute main memory address accessible by software. The register is loaded during system initialization and can only be read by software. The hardware control mask register 208 is an 8-bit register which records machine condition information.

J. INSTRUCTION FORMATS

There are approximately 200 instructions although more or less may be utilized. Each instruction is one of

four different lengths but always an even number of bytes long. Instructions are stored in consecutive storage locations. The address of the leftmost byte is a multiple of 2, and is the address of the instruction.

The eight most significant bits (and in some cases bits 8 through 11 or 12 through 15) of an instruction represent the operation code, while the remaining bits represent one or more operands. An operand may be a register designator, displacement designator, address syllable (logical address) literal value, immediate literal value. The type and number of operands are determined by the instruction format.

III. SYSTEM ORGANIZATION

A. JOB STEP AND TASK

Work to be performed by the computer system is defined externally by a series of job steps via a job control language. A job step is a unit of work to which hardware resources are allocated. Typically a job step consists of several tasks. A task is the smallest unit of user defined work consisting of a stream of instructions executed without parallelism.

B. PROCESS

The user-visible concepts of task and job step are represented in the hardware by a process and process group, respectively. A process is defined as an ordered sequence of instructions which can be executed asynchronously by the CPU (i.e., several processes can be active and sharing resources, but only one process is actually running at any one instant). A process group is a related set of processes necessary to perform one job step.

C. PROCESS CONTROL BLOCK AND SYSTEM BASE

Because processes can relinquish CPU control at various points during their execution, a storage area in main memory is made available to a process to save CPU

status. This status information is utilized to precondition the CPU before a process regains control of the CPU.

The storage area assigned to a process is called a process control block (PCB) 400 on Figure 4. The data contained in a PCB include the address of memory areas (address space) assigned to the process, the contents of all pertinent registers, and the state of the process. Thus a PCB serves as a temporary storage area for information necessary to start or restart a process without any information loss. Each PCB is visible to the hardware and can be addressed by the operating system via a set of hardware tables developed during system initialization and modified during system operation (Figure 5).

There is an absolute main memory area which is referred to as the system base (Figures 5 and 6). This area is developed by firmware and is accessible via the base address register (BAR) 501 which can be read but not written. The system base 502 contains a number of system attributes which include a job step number and a process group number (J, P) respectively for the currently running process. Another attribute in the system base is a pointer to a hardware defined data structure known as the J table 503. This table contains an entry for every job step presently in the system. Each entry in the J table 503 points to an associated P table 504 which is also a hardware defined data structure. This table defines a process group and contains an entry for every process in the process group. Each P-table entry points to a PCB 400.

Referring to Figure 5 the J-table pointer 505 indexed by the J number via computational unit 506 provides access to a J-table entry 503. This entry contains a P-table pointer which when indexed by the P number via computational unit 506 provides access to a P-table entry 504. The P-table entry contains a pointer 507 to the PCB of the current running process. Thus the operating system can access the active PCB using the contents of the BAR 501 and can access any other PCB given its associated (J, P) logic name.

D. MEMORY SEGMENTATION

In a multiprocess environment, such as herein described there are many processes in memory at any given time. These processes vary in size and demand for memory which causes a memory allocated problem. The hardware herein described in cooperation with an operating system (not shown herein) solves the problem by dynamically allocating memory space. Due to the random nature of memory requirements, memory is allocated in variable size segments and the memory allocation can be restructured during process run time. Thus, a process may be allocated a number of noncontiguous memory segments. This memory allocation method is called segmentation.

Segmentation presents an additional problem in that memory addresses have to be modified whenever part or all of a process is relocated. To alleviate this problem the system herein described provides a technique whereby addresses used by a process are logical rather than absolute main memory addresses. These logical addresses are used to develop absolute addresses.

Segmentation also allows each process to access its own or related memory segments via a system of segment descriptors. By accessing a segment descriptor, a process can obtain the address of a segment. Segment descriptors are contained in main memory and are maintained by the operating system.

Each process may have access up to 2068 memory segments. Normally, this would require an equal number of segment descriptors per process. However, since segments can be shared, the operating system groups segment descriptors into segment tables. This grouping is based on accessability by one process (task), a process group (job step), or globally (system wide). Each process may have up to 15 segment tables associated with it. This technique requires only one segment descriptor for each segment which can be accessed by a process via a segment table. Thus, the memory space required for segment descriptors is decreased; memory updating during relocation is reduced; and some program protection

is provided. (The main mechanism for program protection is the ring system. See Patent Application No. ———— filed ———— entitled "Protection of Information in a Multiprogram, Multiprocessor Computer System;" also see Patent Application No. ———— filed ———— entitled "Ring Crossing Mechanism" both assigned to the same assignee as the instant invention.

A process must be able to determine which segments it is allowed to access. Accordingly, the system provides a process with two segment table word arrays (STWA). These arrays contain the addresses of all segment tables accessible to a process. There are two segment table word arrays per process because there are two segment sizes, large and small. Large segments have a maximum size of 2^{22} bytes while small segments have a maximum size of 2^{16} bytes. All segments vary in size in 16-byte increments up to the maximum. A system can typically accommodate up to 28 large segments and 2040 small segments.

Segment table word arrays may be relocated by the operating system; therefore, a process must know the absolute address of its associated STWA's. The PCB for any process contains two words which contain this information which are known as address space words ASW0-1 on Figure 4. Each word points to a segment table word array STWA. The operating system updates the contents of the ASW's whenever the associated STWA's are relocated. Working down the chain of pointers and decoding the segment descriptor is a firmware function and thus once initiated is not visible even to the operating system.

Segmentation defines over 200 million bytes of address space as being available for processes. This number exceeds the capacity of main memory; therefore, a secondary storage (magnetic disk or drum) is used in conjunction with main memory. The operating system creates the illusion that the system has a much larger main memory than is really available. This concept is called virtual memory.

At any given time, a defined segment may or may not be physically in main memory. The contents of a seg-

ment descriptor indicates whether or not the associated segment is in main memory. The hardware detects any attempts by a process to access a segment not in main memory and notifies the operating system. The operating system causes the desired segment to be loaded into main memory from secondary storage. Then the operating system places the segment's memory address in the segment descriptor which is the only place where the absolute address of a segment can be found. This operation is invisible to the process and thus it is not aware that the segment was not in main memory or that it may have to be relocated in main memory. (For details on memory segmentation see Patent Application No. ———— filed ———— and entitled "Segmented Address Development". Also see patent No. ————.)

The computer system herein described provides data and procedure protection by preventing processes from interfering with each other or sharing each other's address space in an unauthorized manner. This protection is accomplished by restricting addressability via memory segmentation and by a ring system.

The segment tables isolate the address space of the various processes in the system. Processes always use a segmented address during execution. A segmented address consists of a segment number and a relative address within the segment (see above referenced application on Segmented Address Development). The hardware checks that the address used by a process is part of the address space assigned to the process. If the address is outside the prescribed address space, an exception occurs. A process cannot refer to data within the address space of another process because the hardware uses the segment tables of the referencing process. Thus, there is no possibility for a process or process group to reference an entity belonging to another process group.

Generally, overlap in address space in the system occurs for those segments shared by all processes. These public segments are created by system programs which check to insure against address conflicts. Thus, segmentation protects user programs against each other and protects the operating system against user programs.

Segments shared by several processes are not protected from misuse by one of these processes. To solve this problem, a ring system is utilized whereby procedure and data segments are grouped into a four-class hierarchy. The four ring classes are numbered 0 through 3. Each ring represents a level of system privilege with level 0 (the innermost ring) having the most privilege and level 3 (the outermost ring) the least. Every procedure in the system has a minimum and a maximum execute ring number assigned to it which specifies who may call the procedure. A procedure is a subroutine which is capable of calling other procedures and passing parameters to them:

The general rules of the ring system are as follows:

1. A procedure in an inner ring has free access to data in an outer ring. Conversely a procedure in an outer ring cannot access data in an inner ring.
2. A procedure in an outer ring can branch to a procedure in an inner ring, but the reverse is not allowed.
3. Each segment containing data is assigned two ring values, one for read (RD) and one for write (WR). These ring values specify the maximum ring value in which a procedure may execute when accessing the data in either the read or write mode.

Each time a procedure instruction is executed, the procedure's ring number (effective address ring, EAR) is checked against the ring numbers assigned to the segment containing the referenced data. The EAR is the maximum number of process ring numbers in the instruction counter and all ring numbers in base registers and data descriptors found in the addressing path. Access to the data will be granted or denied based on a comparison of the ring numbers. For example, if a system table exists in a segment having a maximum read ring value of 3 and a maximum write ring of 1, then a user procedure executing in ring 3 may read the table but may not update the table.

By predesign, rings 0 and 1 are reserved for the operating system and rings 2 and 3 are reserved for the

user. Ring 0 contains those segments critical to total system operation. Ring 1 contains the bulk of the system segments whose failure would not be catastrophic and would allow recovery. The user may utilize ring 2 for checked-out programs and ring 3 for programs being debugged.

F. PROCEDURE CALLS

The procedure call is an important function in the system herein described. Procedure calls are used to pass from one procedure to another, to allow user procedures to employ operating system services; and to achieve a modular structure within the operating system. A procedure call is effected by instructions and a hardware recognized entity called a stack (Figure 7A).

A stack is a mechanism that accepts, stores and allows retrieval of data on a last-in-first-out basis. Stacks reside in special segments called stack segments. A stack segment consists of a number of contiguous parts called stack frames 701 (Figures 7A and 7B) which are dynamically allocated to each procedure. The first stack frame is loaded into the top of the segment and succeeding frames are loaded after it. The last frame loaded is considered the top of the stack. The T-register 702 locates the top of the stack for the currently active process. A virtual T-register exists on the PCB of all other processes in the system.

A stack frame 701 of Figure 7B consists of three areas: a work area 702 in which to store variables, a save area 703 in which to save contents of registers, and a communications area 704 in which to pass parameters between procedures. Prior to a procedure call, the user must specify those registers he wishes saved and he must load into the communications area the parameters to be passed to the called procedure. When the call is made, the hardware saves the contents of the instruction counter IC and specified base registers to facilitate a return from the called procedure.

Each procedure call creates a stack frame within a stack segment 701 and subsequent nested calls create additional frames. Each exit from one of these called

procedures causes a stack frame to be deleted from the stack. Thus, a history of calls is maintained which facilitates orderly returns.

To insure protection between procedures executing in different rings, different stack segments are used. There is one stack segment corresponding to each protection ring per process. A PCB contains three stack base words which point to the start of the stack segments for rings 0, 1 and 2 associated with the process. The ring 3 stack segment can never be entered by an inward call; therefore its stack starting address is not required in the PCB.

IV. PROCESS MANAGEMENT AND SYNCHRONIZATION

The system herein provides for multiprocessing operations which are controlled by an operating system using a combination of software, hardware and firmware. Software creates and deletes processes within the system while hardware and firmware multiplex processes on the CPU. In addition, a combination of software, hardware and firmware provide for synchronization between processes.

Processes are normally, but not always, started and stopped at the initiation and termination of i/o operations, during related job handling, and at other times for purposes deemed necessary by the operating system. Therefore, a comunications system is necessary to efficiently start and stop related processes and to pass information between them. The hardware system herein provides internal messages called semaphores to provide a communications link between the processes.

A. PROCESS STATES

A process can be in one of four possible states at any time: running, ready, waiting or suspended. The hardware recognizes these four possible process states and executes various firmware procedures to effect process dispatching, state changes and to maintain data structures based on a process's state. The PCB contains a

state field which defines the current state of its associated process.

A process is in the running state when it has control of the CPU. This state involves supplying the CPU with an address space (segment tables) and a starting address. The CPU then executes instructions in the procedure segments of the process. The process name J table word (logical address) of the PCB for the currently running process is retained in the running process word (BAR +60) within the system base (Figure 6). (Note: The system base shown in Fig. 5 is the same as that shown in Fig. 6, but with some details omitted.)

The ready state is equivalent to running state except that the process does not have control of the CPU because it has not been recognized by the CPU. A process in the ready state is in contention for the CPU with other ready processes and the running process.

A process is in the wait state when it cannot continue until a specific event occurs such as a message via a semaphore. A waiting process is not in contention for the CPU but it may be in contention with other waiting processes for the required event.

A suspended process is a process which has been stopped for a time by software and may be resumed later. The decision to stop and resume the process is external to the process. Thus, a suspended process is not active and therefore cannot receive notification of event occurrences and cannot utilize the CPU.

A process is suspended under the following conditions:

(1) by executing a Terminate instruction (as a result of having completed all its functions.)

(2) By execution of a Suspend instruction by the operating system.

(3) By the occurrence of an exception condition whereby control is transferred to the operating system.

B. PROCESS DISPATCHING

Processes move from one state to another voluntarily by action of the process while running or involuntarily by the actions of other processes. CPU firmware, known

as the dispatcher, controls the transaction of processes between states. The dispatcher uses a set of queues (to be later described) to manipulate processes which are in the ready or the waiting states. Suspended processes are controlled by software.

Referring to Figures 6, 8 and 9, a ready or waiting process is represented by a PCB and a special queue entry called a process link. Figure 9 shows an exploded view of contents of the GO segment 802, and contains process links 803a-803b and 803c-803g of active processes, and free process links 805a-805c of suspended processes. Each process link specifies the process name (J, P), the process priority and a pointer to the next process link in the queue. There are various types of queues such as wait queue 803a-b and ready queue 803c-g.

A hardware device similar to the J table, known as the G table, (Figures 6 and 8) contains pointers to all general (known system wide) segments 802-802n. The first element, GO, of the G table 801 points to that segment 802 containing the dispatcher queues. A G-table pointer to the G table 801 is found in the system base 502 on Figure 5. Also in the system base is an entry called the internal process queue word (IPQW) which identifies the head 805 of the ready queue 803c-803g in the GO segment 802.

Thus, the dispatcher can examine all ready processes by consulting the ready queue 803c-803g. When the currently running process changes states, the dispatcher removes the process link at the head of the ready queue and uses the J, P name to access its PCB. The process defined by the PCB then becomes the new running process.

Since more than one process may be awaiting on the same event, a queue of waiting processes 803a-803b exists for each event. Waiting processes are also strung together via process links 805 residing in the GO segment. A pointer to the head of a wait queue exists in a semaphore 903 (to be later described). A number of events exist for which a process may wait; therefore, there are a number of wait queues each of which has an associated semaphore 903, 904.

The number of processes ready or waiting varies dynamically. Thus, the number of process links required for the ready and wait queues also varies. This fact introduces a mmeory management problem for the dispatcher The problem is solved by another queue called the free process link queue 805a-c. This queue links together all process links is segment GO that are not being used by the ready or the wait queues and can be used to extend a particular queue of ready or waiting processes. A pointer 901 to the head 902 of the free process link queue 805 resides near the beginning of the GO segment 802.

C. PROCESS SYNCHRONIZATION

Process synchronization is required to coordinate the activities of two processes working on the same task. The synchronization is achieved using semaphores 903-904 which are data structures residing in the address space of communicating processes. A semaphore is used to signal event occurrence and to handle queues of messages. An event in this context is anything observed by a process which may be of interest to some other process. The event may be the completion of an asynchronous operation or the availability of a resource.

A process uses two semaphore operations to signal an event occurrence. One operation sends a signal to a semaphore; the other picks up a signal from a semaphore. (The sending operation is often called a V-operation; the receiving operation is called a P-operation). The sending operation allows a process to send data or a signal that data are ready. The semaphore stores the signal until another process is ready to pick it up. Thus, the sending process is free to proceed, since it has sent the data. The receiving operation examines a specified semaphore and picks up the signal. If a signal is present, the receiving process continues executing. However, if there is no signal at the semaphore, the receiving process enters the wait state. The semaphore then serves as a pointer to the head of a wait queue. The process remains in the wait state queued at the semaphore until another process sends a signal to that particular semaphore.

Thus, a semaphore can hold a signal until a process picks it up, or a semaphore can hold a process until a signal is sent to it.

Messages can also be passed from process to process. A message has the same present or not present quality as a signal plus additional information. Part of the information is supplied by hardware and part is supplied by the procedure of the process that sent the message. A message carries the process name of the sending process. Thus, many processes can send information through a single semaphore stamped with the sender's name.

A message semaphore may have a queue of messages waiting to be picked up by processes. As with signal semaphores, requirements for memory space increases and decreases thus presenting a memory management problem. Again, the problem is solved with a queue of free message links. These links reside in a known place in a segment that can easily be found when needed to supply or absorb message links.

Because semaphores and the queues built on them are shared by different processes, the total semaphore structure is protected. This is accomplished by hardware and software conventions that restrict access to any segment containing semaphores. Thus, semaphores must be in semaphore descriptor segments, some of which may be G segments (if system communications is necessary). However, all G segments (except GO) are semaphore descriptor segments.

Each semaphore descriptor contains a pointer to a semaphore. Semaphore addresses are developed via a semaphore descriptor, thus providing added protection for the semaphore. A semaphore segment can be addressed logically using a segment number and a relative location within the segment or directly using the G, D number.

PROCESS CONTROL BLOCK STRUCTURES

Referring to Figure 4 there is shown the format of the process control block (PCB). The process control block 400 is a storage area in main memory made available to a

process to save the CPU status. Addressing a PCB is performed as described supra in relation with Figure 5. The PCB pointer 507 (Figure 5) points to the process control block PCB at memory location 0 on Figure 4. It will be noted that proceeding in a downward direction memory locations increase by 4 bytes whereas in proceeding in an upward direction from memory location 0 they increase by 8 bytes. The downward memory locations are considered positive from 0 whereas the locations in an upward direction from 0 are considered negative directions. The upward locations are optional and may or may not be included in the process control block; also locations 148 through 176 are also optional. (Note that the numerals under memory location specify the displacement in bytes from the 0 reference location of the process control block PCB and are not to be confused with the reference numerals commonly used to identify parts in a patent drawing). Starting at byte 0 up to but not including byte 16 there are stored four process main words PMW 0 through PMW 3 with each process main word PMW being four bytes in length. Process main word 0 occupies bytes 0 through 3 and is comprised of 4 parts: a capability byte, a priority byte, a state byte and a decor extension byte DEXT. Referring to Figures 10a through 10d there are shown details of process main word PMW 0, with further details of the capability byte 1001 shown on Figure 10b. Referring to Figure 10b, the first bit 1005 is the accounting mode bit for indicating whether or not time accounting functions are being performed for the process. When the accounting mode bit 1005 is set to binary 0 no time accounting function is being performed for the process; whereas when the accounting mode 1005 is set to binary 1, time accounting is being performed. The scientific mode bit 1006, when set to zero, indicates that saving of scientific register of the machine is not performed and the scientific register saving area located at bytes 148 to 176 on Figure 4 does not exist in the process control block PCB. When the scientific mode bit 1006, is set to binary 1, the scientific optional feature exists and is being used in the process, and the scientific registers savings area is used to save the contents of the scientific registers when necessary.

The code mode bit 1007 indicates whether or not a standard code set or compatibility code set is being used by the process, with a binary 0 in that position indicating that standard code set is being used; whereas a binary 1 in the third bit position 1007 indicates a compatibility code set is being used. The remaining of the bits of the capability byte are set to zero.

Details of the priority byte 1002 are shown on Figure 10c. Referring to Figure 10c the first four bits 1008 of priority byte 1002 is utilized to set the priority level of the process associated with that given process control block PCB. Each process is assigned one of 16 levels of priority which is used for ordering competing processes i.e. (a) for choosing the process to be run among ready processes, (b) for putting processes in queues. Priorities decrease from 0 to 15, and for a given priority level the FIFO (first in first out) rule is applied. The next 4 bits 1009 of priority byte 1002 are zeroes.

Referring to Figure 10d details of the state byte 1003 are shown. A state byte is utilized to provide information with regard to the process associated with the process control block PCB 400. The active field bit A 1010 is set to binary 1 when the process is activated. The suspend field S 1011 is set to binary 1 when the process is suspended. The substate field SS 1012 is a 2 bit field and defines the following substates of the process: (a) when set to binary 00 the process is inactive; (b) when set to binary 01 the process is waiting in the queue of ready process (Q/PR/RDY); (c) when set to binary 10 the process is waiting on a semaphore in a queue of semaphores (Q/PR/S); (d) when set to binary 11 the process is being executed by the processor. The mid-operation field (MOI) 1013 is set to binary 1 when an interrupt happens and is taken care of during the execution of an instruction—i.e. before the completion of the process. The extended decor mode bit EXTD 1014 is set to 1 when the process is operated in an extended decor mode which is an emulation mode of the machine. Bits 1015 and 1016 are set to 0. The fourth byte of process main word PMW 0 contains the decor extension number and is utilized when the system is in emulation mode.

Process main word PMW 1 is stored in bytes 4-7 of the process control block PCB. Details of PMW 1 is shown on Figure 10e. The status byte 1016 is the first byte in PMW 1 and stores the status register contents. The multiprocessor byte MP 1018 is significant in a multiprocessor architecture otherwise this field is zero. The second and fourth bytes of process main word 1 are the MBZ fields 1017 and 1019 respectively which must be zero for normal operation.

Process main word PMW 2 occupies bytes 8 through 11 of the process control block and is shown in more detail on Figure 10f. Referring to Figure 10f the field from bit 4 through bit 31 contains the local name SEG, SRA 1021 of the semaphore to which the PCB is linked when the process is either in the waiting or suspended states. The exception class and type field 1023 contains the class and the type of the interrupt-like exception which cause the process to enter the suspended state after an exception. The field from bits 4 through 15 is meaningless 1022 when a process is in a different state than those mentioned above.

Process main word PMW 3 occupies bytes 12 through 15 in PCB 400 and points to a decor extension table. Referring to Figure 10g for details of PMW 3 the DETSZ field 1024 defines the number of entries in the table and if this field is zero no decor extension is allowed to the process. The DETA field 1025 is the absolute address of the decor extension table in units of 16 byes and is significant only if DETSZ is not 0. The decor extension table is made up of DETSZ entries. Each entry is one byte size. The DEXTth entry of the table defines the capability of the process to operate in the decor extension mode DEXT. When the DEXTth byte is 0 the decor extension number DEXT is not allowed, whereas if the DEXTth byte is 1 the decor extension number DEXT is allowed. Values of DEXT other than 0 and 1 are illegal. (See Figures 10a DEXT number 1004).

Bytes 16 through 23 of PCB 400 contains 2 address space words ASW 0 and ASW 1 respectively and each ASW contains a pointer to an array of segment table words. Both ASW 0 and ASW 1 respectively have the

same format shown on Figure 10h. The size of the array of the segment table words is defined by the number of segment table words in an array and typically comprises six for ASWO and eight for ASW 1. The STWSZ field 1026 indicates the size of the array of the segment table words. The segment table word array field STWA 1027 contains the absolute address STWA of the array in units of 16 bytes—i.e. the absolute address of the array is 16 times STWA in bytes.

Bytes 24 through 27 in the PCB contain an exception word EXW shown in greater detail on Figure 10i. The exception word contains a pointer (SEG, SRA) 1029 to an exception class table which defines the action to be taken following a process exception according to its class as stored in process main word PMW 2. (See Figure 10f). The MBZ field 1028 of exception word EXW must be 0.

The stack word SKW located in bytes 28 through 31 of the PCB contains the value of the top of the T register of the stack of the process when the process is not running and is shown in greater detail in Figure 10j. Referring to Figure 10j, bits 0 and 1 define the TAG field 1030. The TAG indicates the type of descriptor by its contents and must be zero for SKW. Bits 2 and 3 of the SKW word contain the RING field 1031 which contains the ring number associated with the segmented address of the stack for protection purposes and in this case must be zero. Bits 4 through 31 contain the segment number SEG, and the segment relative address SRA 1032 and is a field which identifies the segment described in a segment table and the segment relative address within the segment. The stack word SKW is updated every time the process leaves the running state. It is used to restore the T register contents every time the process becomes running. In this last case the TAG 1030 and RING 1031 are tested to be zero, otherwise an illegal PCB exception occurs.

Bytes 32 through 35 of the PCB 400 contain the instruction counter content word ICW sometimes also referred to as ICC. Referring to Figure 10k there are shown details of the instruction counter word ICW wherein the TAG field 1033 must contain binary 00 (i.e.

values other than zero are illegal in the instruction counter). The current RING field 1034 occupying bits 2 and 3 defines the current ring number of the process to be used in determination of access rights to main storage. Bits 4 through 31 define the segment number and the segment relative address (SEG, SRA) 1035 which define the address of the next instruction to be executed.

The MBZ field in bytes 36 through 39 must be zero. (Note the MBZ field always indicates a field which must be zero). The MBZ word is tested every time the PCB is accessed from the name J, P. If it is not zero an illegal PCB exception occurs.

Stack base words SBW 0-2 occupy bytes 40-51 in the process control block 400. These words have the same format which is shown in greater detail on Figure 101. They are utilized during stack operations and whenever used their TAG field 1036 and RING field 1037 must be zero otherwise an illegal PCB exception occurs. Bits 4 through 31 contain the segmented address (SEG, SRA) 1038 of the first bytes of the stack segments for ring zero, 1 and 2 respectively.

Bytes 52 through 83 of the process control block 400 is a space reserved for the base registers saving area (8 words). Bytes 84 through 147 is a saving area which is utilized to save the values of all general registers (16 words). Bytes 148 through 179 is a saving area which is utilized to save the scientific registers (8 words).

Five double words are provided in the PCB 400 above the PCB zero address, for time accounting purposes when the accounting mode bit in the PMW 0 word is set. These words are located from PCB address minus 8 to PCB address minus 40. Each word contains a time or a time interval expressed in microsecond units in its first 52 bits with bits 52-63 filled with zeroes. The residual time out double word RTO (first 8 bytes above 0 in the PCB) contains the quantum of time which is actually spent by the processor on behalf of the process before a time out exception occurs. The RTO word is updated in the following way: each time the process exits the running state the process timer value is stored in the RTO word. Each time the process enters the running state, the process timer value is loaded from the RTO.

The running time accounting RUA double word at bytes 7 through 15 is a time counter which specifies the total amount of processor time a process was in the running state. The time accounted for is the time actually spent by the processor on behalf of the process exclusively. The RUA word is updated in the following way: each time the process exits the running state, the value of the process timer PT is read. The difference of the contents of RTO and PT is added to RUA. (Consecutively, the PT value is stored in RTO). Note that the time during which the process is suspended is not computed. The RTO and RUA words are updated even if the accounting mode bit is set to 0. However the CET, RTA, and WTA words (to be later described) are provided in the process control block only if the accounting mode bit in the process main word PMW 0 is set to 1. They are updated only in this case.

The waiting time accounting WTA word at bytes 17 through 23 is a real time counter which specifies the total amount of real time the process was in the waiting state. The WTA word is updated in the following way: each time the process exits the waiting state the time of day clock (not shown) value TOD is read and the Value of TOD minus the value of CET word is added to the WTA word.

The ready time accounting RTA word located at bytes 24 through 31 is a double word which is a real time counter which specifies the total amount of real time the process was in the ready state. The RTA is updated in the following way: each time the process exits the ready state, the time of day clock value TOD is read, and the contents of TOD minus the contents of CET is added to RTA.

The current entry time CET double word at bytes 32 through 39 contains the time of day at which the process entered one of the following states: ready, waiting, running, and suspended.

SYSTEM BASE STRUCTURES:

Referring to Figure 6 the format of the system base 600 is shown. The system base resides in absolute main memory and is developed by firmware and is accessible

via the boundary address register (BAR) which can be read but not written. The boundary address register BAR is below an area in main memory reserved for hardware and separates this area in memory reserved for hardware and the system base 600. Referring now to Figure 6 the system base 600 contains a number of system attributes which includes a job step number and a process group number (J, P) for the currently running process. From the logical name of the process J, P, the absolute address of the corresponding process control block PCB is obtained. The size and address of the J table are defined by the contents of the J table word (JTW). This word is located at the address defined by the BAR register. The format of the JTW is shown on Figure 11a. The size (JTSZ) 1101 or the J table 1204 on Figure 12 defines the number of entries in the J table 1204 which may be up to 255 entries. The JTSZ 1101 is an 8 bit positive integer; an out of J table exception occurs if J is greater than JTSZ. The absolute address of the J table 1204 is obtained by multiplying the J table pointer 1102 by 16. The J table 1204 contains J table entries whose format is shown in greater detail on Figure 11b. Each J table entry defines the absolute address of a P table 1205 which is obtained by multiplying the P table pointer 1104 by 16. The size (PTSZ) 1103 of a P table defines the number of entries in the P table. The PTSZ is an 8 bit positive integer which may typically vary from 0 to 255 to indicate the number of entries in the P table. An out of P table exception occurs if P is greater than PTSZ. Each entry of the P table 1205 defines the absolute address of a process control block (PCB) 1206 by multiplying the process control block pointer 1107 by 16. A presence indicator P 1105 indicates the absence of a PCB 1206 when set to binary 0 and indicates the presence of a PCB when set to binary 1. (When the presence indicator P 1105 is found to be 0 a vacant P table entry exception occurs). Bits 1 through 7 of the P table indicator (Figure 11c) must be 0 (MBZ) 1106, otherwise an illegal P table entry exception occurs.

At address BAR plus 4 of the system base 600 there is the format byte of a G table word (GTW) shown in

greater detail on Figure 11d. The size and the address of a G segment-table 1212 on Figure 1200 are defined by the contents of the G table word (GTW). The size (GTSZ) 1108 of the G table 1212 defines the number of entries in the G table which may typically be up to 255 entries. GTSZ is an 8 bit positive integer; an out of G table exception occurs if the G number is greater than the GTSZ. The absolute address of the G table 1212 is obtained by multiplying the G table pointer 1109 by 16. The format of the G segment table entry has a two word size (8 bytes) and is called a G segment descriptor. The format of the G segment descriptor is shown in detail on Figures 11e and 11f. All G segment descriptors are direct and therefore the indirect bit I, 1111 must be 0 otherwise an illegal segment descriptor exception occurs. The presence indicator P 1110 is a one bit field which when set to binary 1 indicates that a segment is defined in main storage for the segment number to which that descriptor corresponds; whereas if it cleared to 0 no segment is defined and a reference to the segment descriptor causes a missing segment exception. The available bit A 1112 is a one bit field which indicates whether or not the segment is available; it is only checked if this segment is defined (i.e. P equals binary 1), otherwise it is ignored. The used flag field U 1113 indicates whether or not the segment has been accessed. If the U bit is set to binary 0 the segment has not been accessed; whereas if the U field is set to binary 1 the segment has been accessed. The written flag field W 1114 indicates whether the segment has been written. If W is set to binary 0 the segment has not been written; whereas if W is set to binary 1 the segment has been written. The gating indicator GS 1115 of a G segment descriptor must be set to binary 01, otherwise an illegal segment descriptor exception occurs. The reason for this is that a G segment always contains semaphores (although the reverse is not true i.e. all semaphores are not required to be in a G segment) and instructions on semaphores require the GS code to be binary 01. The absolute address of the base of a segment 1214 is defined in the G segment descriptor of Figure 11e by the 24 bit base field 1116; the content

of this field is multiplied by 16 to obtain the absolute address. The second word of the G segment descriptor of Figure 11f occupies bit position 32 through 63 in the G table 1212. The RSU field 1117, bits 32 through 39 is reserved for software use and is generally ignored when used as a G segment descriptor as it is in this case. The MBZ field 1118 must be 0 otherwise an illegal segment exception occurs. Since the MBZ field 1118 occupies bits 40 through 51 it sets the SIZE field 1119 which is the field for a small segment SIZN; hence all G segments must be of the small segment type. The segment SIZN 1119 is a 12 bit positive integer defining the number of bytes in the segment and the segment size is interpreted as a multiple of 16. Therefore the segment size for a G segment 1214 cannot exceed 2^{16} bytes (small segments).

Referring once again to the system base 600 of Figure 6 there are 9 system exception cell words located between BAR plus 8 and BAR plus 44. The format of the system exception cell words EXC is shown on Figure 11g. Since semaphores are utilized for transmitting messages to dedicated processes when a system exception occurs the pointers to these semaphores are found in 9 locations of memory each location called a system exception cell—one per class of system exception. The MBZ field 1120 must be set to binary 0 otherwise a system check occurs. Each exception cell (EXC) contains the system name G, D 1121 and 1122 respectively.

The channel exception cell located in BAR plus 44 of the system base 600 has a format which is similar to the system exception cell previously discussed and contains the system name GD of a semaphore which is used for transmitting messages to dedicated processes when a channel exception occurs.

An internal processor queue word IPQW is located beginning at BAR plus 48 and details of its format are shown on Figure 11h. The IPQW word points to the head of a queue of processes ready (Q/PR/RDY) as shown on Figure 9 by reference numerals 905 and 805. The queue of processes ready (Q/PR/RDY) links all processes which are in the ready state. It is referenced by the HEAD

OF Q/PR/RDY-field 1124 (Figure 11h) of the IPQW word by pointing to the top of the ready process queue. The HEAD of Q/PR/RDY-field 1124 contains a 16 bit positive integer which is the displacement from the base of the G segment number 0, referred to as the GO segment, to the first byte of Q/PR/RDY. If this Q/PR/RDY bit field is 0, the ready queue is considered to be emtpy. The MBZ field 1123 must be 0 otherwise a system check occurs.

At BAR plus 52 of the system base 600 there is shown the storage for the initial and current retry counts whose format is shown in detail on Figure 11i. The NFS field 1125 is a nonfunctional storage field and is not utilized by the system base. The initial retry count field 1126 and the current retry count field 1127 are used to control the number of times automatic instruction retry is executed before a machine error is made to produce a machine failure exception condition. They are loaded with the same number by a Reset Retry Count (not shown herein).

The running process word (RPW), shown in Figure 11j, is stored in BAR plus 56 of the system base 600 and is used to store the name of the running process with its priority in case of a monoprocessor architecture. The NFS fields 1128 and 1131 respectively are nonfunctional storage fields and may be utilized for any purpose by any facility but is generally not utilized by the system base. The priority level of a running process is stored in the PRI field 1129. An asynchronous trap bit is stored in AB field 1130; whereas an asynchronous trap ring is stored in ARN field 1132. The logical name J, P of the running process in case of a monoprocessor architecture is stored in the J, P field 1133.

An Absolutization Table Pointer word shown on Figure 11k is located at BAR plus 60 in the system base 600 and is utilized in initial system load to initialize the absolute addresses in the initial system load (ISL) program by adding the contents of BAR to all absolute addresses in the ISL program. The Absolutization Table Pointer 1135 defines the location of an Absolutization Table (not shown). The Absolutization Table Size is shown by the ATSZ field 1134.

The CPU serial number word shown on Figure 11l is a 4 byte word located at BAR plus 64 and contains the serial number of the CPU in the CPU serial number field 1136.

A main storage upper limit word shown on Figure 11m is located at BAR plus 68 and indicates the main storage upper limit 1139 by providing the absolute address of the last available word in main storage.

At BAR plus 72 there is located a word shown on Figure 11n for providing the initial system load ISL device channel number (CN) 1140 and the hardware device channel number (CN) 1141.

The type and subtype of a device used in the computer system is shown by a hardware device type word (Figure 11o) in fields 1143 and 1144 respectively; where the RSU field 1142 is reserved for software. This word is found in the system base at BAR plus 76. A similar word having a similar type format shown on Figure 11p contains the type and subtype of the device used in the initial system load. This word is located at BAR plus 80.

When the restart button of a computer is pressed, a simulated V-operation is performed on a semaphore and the Ready state is entered. A pointer to this semaphore is found at BAR plus 84 of the system base 600 and is called a restart cell word, and has a format shown on Figure 11q. The format is similar to the system exception cell described supra and contains the system name G, D of a semaphore in the G field 1149 and D field 1150 respectively. The MBZ field 1148 must be 0.

When there is more than one processor to the computer system, a word is provided in the system base 600 at BAR plus 88 for multiprocess extension. Details of this word are shown on Figure 11r.

Examples of System Base and Process Control Block Use

Referring to Figure 12 there is shown one example, how the system base may be utilized in combination with the process control block in order to address and access a user segment, a system segment, or a queue of processes read (Q/PR/RDY) segment. Main memory 1200 has a

portion 1203 reserved for hardware use. A boundary address register BAR 1202 separates the system base 1215 from the portion of memory 1203 reserved for hardware. The boundary address register BAR 1202 is utilized to address items in the system base 1215 by adding the contents of the boundary address register to the displacement in 4 byte units of the item desired in the system base. This address then points to the first byte of the item in the system base desired. In Figure 12 the BAR 1202 is pointing at the J table word (JTW). The J table word, as previously discussed, has a pointer which points to a J table 1204. By indexing to the J number shown on Figure 5, a J table entry 1216 is obtained. At the J table entry there is a P table pointer which points to the absolute address of the P table 1205. By indexing to the P number (see Figure 5) within P table 1205 the absolute address of the process control block 1206 is obtained. As previously shown in process control block PCB 1206 there are two address space words ASW 0 and ASW 1. The high order bits of the segment table number field STN in the base register 1201 is used to access one of these two address space words, in this instance ASW 1 which has a segment table word array STWA pointer that points to segment table word array STWA 1208. Together with the segment table number STN of the base register 1201 one of 8 segment table words is accessed in STWA 1208, which points to one of 8 segment tables 1210. The segment table entry STE from base register 1201 is then utilized to make one of 256 entries in segment table 1210 where a segment descriptor is located. The segment descriptor is then utilized to access a user segment 1211. (For greater detail see Application No. ———— filed on ———— (entitled Segmented Address Development and assigned to the same assignee as the instant invention and is hereby incorporated by reference herein).

In order to access a system segment 1214 which is utilized to store semaphores a G table word GTW is utilized in the system base 1215. The address of the G table word is obtained by adding the displacement of the G table word in the system base to the boundary

address register BAR 1202. (See Figure 6). The G table word GTW includes a G table pointer which points to a G table 1212. By utilizing a G number available to the system and indexing in the G table a G segment descriptor is accessed which is utilized to address a system segment 1214.

Similarly the system base 1215 is utilized to access the queue of process ready (Q/PR/RDY) 1213 by locating an internal processor queue word IPQW which points to the Q/PR/RDY segment 1213.

Control Unit

Referring to Figures 13a-13c details of the control unit are shown. The control unit, although shown separate from the central processing unit (CPU), is in actuality a part of the CPU and is comprised of a control store unit CSU 1301, a control store interface adapter CIA 1302 and appurtenant subunits, control store loader CSL 1303 and control and load unit CLU 1304.

The control store unit CSU 1301 receives microinstructions from the control store loader CSL 1303 via the control and load unit CLU 1304 and the control store interface adapter CIA 1302. Under normal operating conditions, microprograms are loaded from an external source during system initialization and become a permanent control function of the machine. However the control store unit CSU 1301 has the ability to be reloaded and initialized in a manner that provides for a variety of central processing unit CPU 1306 operational modes. The following modes of operation of the CPU are available under control of the CSU 1301; (a) native mode; (b) emulation mode; (c) concurrent native and emulation modes; (d) diagnostic mode. This capability is possible because the micro-instructions resident in the CSU are the source of micro-operations used to control the operation of all other CPU functional units such as the emulation unit 1316, the arithmetic logic unit ALU 1317, the instruction fetch unit IFU 1318, the address control unit ACU 1319 and the data management unit DMU 1321. Also shown within the central processing unit CPU 1306 are previously described general regis-

ters 1307, base registers 1308, scientific registers 1309, T-registers 1310, status registers 1311, instruction counter IC 1312, and hardware control mask register 1313.

Typically the control store unit CSU 1301 is a 9K bipolar integrated circuit programmable read-only memory (PROM) mixed with read/write random access store (RAM). It has a typical 150 nanosecond read cycle and a 450 nanosecond write cycle. Each location of control store stores one 84-bit micro-instruction word (to be later more fully described), and each micro-instruction word controls one CPU cycle. As each location of the control store of the control store unit CSU 1301 is read, its contents are decoded by micro-operation decoders which provide micro-operation control signals each of which causes a specific operation within the CPU to take place (to be later described in detail).

By grouping locations within each micro-instruction word (to be later described in detail) control store sequences are obtained that can perform a specific CPU operation or instruction. As each instruction is initiated by the CPU, certain bits within the op-code are used to determine the control store starting sequence. Testing of certain flops (not shown) which are set or reset by instruction decode functions allows the control store memory to branch to a more specific sequence when necessary.

The control store interface adapter CIA 1302 communicates with the control store unit 1301, the data management unit DMU 1321, the address control unit ACU 1319, and the arithmetic logic unit ALU 1317 for directing the operation of the control store memory 1333 of Figure 13b. The CIA 1302 includes logic for control store address modification, testing, error checking, and hardware address generation. Hardware address generation is utilized generally for developing the starting address of error sequences or for the initialization sequence.

The data management unit DMU 1321 provides the interface between the CPU 1306 and the main memory and/or buffer store memory shown on Figure 1. It is the responsibility of the data management unit to recognize which unit contains the information required by other units and strobe the information into the CPU

registers at the proper time. The data management unit DMU also performs the masking during partial write operations.

The instruction fetch unit IFU 1318 interfaces with the DMU 1321, the ACU 1319, the ALU 1317, and the CSU 1301, and is responsible for keeping the CPU supplied with instructions. The instruction fetch unit has the next instruction available in its registers before the completion of the present instruction. To provide this capability, the instruction fetch unit IFU 1318 contains a 12-byte instruction register (not shown) that normally contains more than one instruction. In addition, the IFU, under control of the CSU, requests information (instructions) from main memory before the instruction is actually needed, thus keeping its 12-byte instruction register constantly updated. Instructions are thus prefetched by means of normally unused memory cycles. The instruction fetch unit also decodes each instruction and informs the other units of the instruction's length and format.

The address control unit ACU 1319 communicates with the IFU, ALU, DMU, and the CSU via the CIA. The ACU 1319 is responsible for all address development in the CPU. All operations of the ACU, including transfers to, from, and within the unit, are directed by CSU microoperation and logic in the unit. The normal cycling of the ACU depends on the types of addresses in the instruction rather than on the type of the instruction. Depending on the address types the ACU may perform different operations for each address in an instruction. The ACU also contains an associative memory 1319a that typically stores the base address of the 8 most recently used memory segments, along with their segment numbers. Each time a memory request is made, the segment number is checked against the associative memory contents to determine if the base address of the segment has already been developed and stored. If the base address is contained in the associative memory 1319a, this address is used in the absolute address development, and a considerable amount of time is saved. If the base address is not contained in the associative memory 1319a it is developed

by accessing the main memory tables. However, after the base address of the segment is developed, it is stored in the associative memory, along with the segment number, for future reference.

Interfacing with the ACU, IFU, DMU, and the CSU is the arithmetic and logic unit ALU 1317. Its primary function is to perform the arithmetic operations and data manipulations required of the CPU. The operations of the arithmetic logic unit are completely dependent on micro-operation control signals from the control store unit CSU 1301.

Associated with the ALU 1317 and the CSU 1301 is the scratch pad memory unit LSU 1315, (sometimes referred to also as the local store unit). It is typically comprised of 256-location (32 bits per location) solid state memory and selection and read/write logic for that memory. The scratch pad memory 1315 is used to store CPU control information and maintainability information. In addition, the scratch pad memory 1315 contains working locations which are primarily used for temporary storage of operands and partial results during data manipulation. Also associated with the ALU 1317 is an auxiliary memory 1317a comprised typically of 64 flip-flops for storing miscellaneous states of the computer system.

The CPU also has a clocking unit 1320 and is essentially 2 clocking systems in 1: the first clocking system generates the timing for the control interface adapter CIA 1302 and the second clocking system generates the timing pulses for the operations of the functional unit within the central processing unit.

Referring now to Figure 13c there is shown the format of the control store word 1325. The control store word is typically 84 bits wide and is divided into 6 main fields:

a. sequence type field 1326 (3 bits);

b. branching and/or micro-operations 1327 (23 bits);

c. constant generation and designation 1328 (14 bits);

d. data to bus 1329 (8 bits);

e. micro-operations 1330 (32 bits); and

f. checking 1331 (4 bits).

The 3-bit E field of the control store word 1325 is used as a sequence control field. There are typically 7 different sequence types and 1 reserved type for the instant computer system. Referring to block 1335 of Figure 13b, when E field equals binary 0, 1, or 2, the branching field A, B, C, D, and L of micro-instruction 1325 is utilized to generate the next address. The first 6 bits of KS register 1337 is utilized together with the B field, a C test results, the D test results and the L field to provide the next address of the next micro-instruction which is then placed in address register KS 1337. When the E field is set to binary 4 (see block 1335) the next address selected is taken from interrupt return register KA 1339. The address stored in the KA register is the one generated by the next address generation logic when the hardware interrupt occurs. When the E field is set to binary 5 a branch is used to initiate a subreturn from a microprogram subroutine. When used, the contents of the return register KR 1346 are used as the next control store address. The return register 1346 is loaded by issuing a control store command which will load present control store address in KS register 1337 plus 1, from incrementor 1338, into the KR register 1346. A one-level-nesting subroutine ability is provided via the KT return branch register 1347. Every time the KR register 1346 is loaded the old contents of the KR register is transferred to the KT register 1347 every time the microprogram return is called; the contents of the KT register will transfer to the KR register. Third level nesting subroutine ability is provided by the KU register 1340; and fourth level nesting subroutine ability is provided by the KV return branch register 1349. When the E field of the control store word is set to binary 6 the next control store word addressed is equal to the present address in KS register 1337 plus 1 in incrementor 1338. When the E field is set to binary 7 the CSU 1301 enters the diagnostic mode and the next address will be the present address plus 1.

In addition to the sequencing control of branching to the next control store address described above and shown in block 1335, there is hardware generated se-

quence control shown in block 1336 of Figure 13b. (Note: Blocks 1335 and 1336 are in actuality hardware registers drawn so as to depict the different forms that the micro-instruction words may take.) The hardware generated branches are overriding conditions (such as errors, initialize, control store scan, etc.) which surpass the E field and force a fixed address into the control store address register KS 1337. The branch is made by forcing an interrupt line high (not shown) for one clock period and storing the address which would have been generated under the control of the E field into the KA interrupt return register 1339. A hardware generated address will be placed into the control store address register. Certain hardware/firmware generated interrupts take priority when the interrupt-block flip-flop (not shown) which prevents additional interrupts in their class from being executed until the interrupting condition has been satisfied. A firmware micro-operation exists for controlling the resetting of the interrupt-block flip-flop for those sequences which are under firmware control. Those sequences under hardware control automatically generate a reset of the block-flop at the end of the sequence. The following conditions, listed by priority, exists in this category; (a) control store load; (b) control store scan; (c) hardware error; (d) software error. The remaining hardware conditions do not set the interrupt block-flop but do cause an immediate action to occur when generated. The following conditions listed by priority, exist in this category;

(a) initialize;

(b) soft-clear;

(c) enter maintenance panel;

(d) enter maintenance panel;

(e) hardware exit.

An initialize signal causes the CSU 1301 to branch to address binary 0, clear hardware resettable errors and execute a control store load operation followed by a control store scan sequence under hardware control. It will also perform system initialize. A soft-clear signal causes

the CSU 1301 to branch to address binary 0, clear hardware resettable errors and reset the interrupt block-flop. An enter maintenance panel signal causes the CSU to branch to the address preset in the CSU address switches on the maintenance panel (not shown).

An enter maintenance channel signal causes the CSU to branch to an address generated via the maintenance channel (not shown). The address loaded is from maintenance bus QMB 1344, which is part of the maintenance channel, and is right-justified. A hardware exit signal causes the CSU to branch to binary address 2. This sequence is used as a maintenance facility. At the end of the sequence a return is initiated by issuing an E field branch with the E field set to binary 4.

A control store load signal causes the CSU to branch to address binary 0. It also turns off the CSU read-cycle flop (not shown), the system clock 1320, and places the CSU in the load state. In the load state the CSU can be loaded from the control store loader CSL 1303, the IOC 1305, the main memory 102, or the maintenance panel 1355. When loaded from the CSL an automatic scan is generated at the end of the load. When loaded from any other media a scan may be issued by either generating a micro-operation signal or setting the scan switch on the maintenance panel. A control store scan signal causes the CSU to branch to an address binary 0. A control store scan is under hardware control for the duration of the sequence. During the scan the system clock 1320 is off and therefore no commands or tests are executed. At the end of the scan sequence the hardware transfers the contents of the interrupt return register KA to the address register KS, the system clock is turned on and control is returned to the firmware.

A hardware error signal causes the CSU to branch to address binary 4. In the normal processing mode a hardware error detected in any CPU functional unit will activate a hardware error line (not shown). The control store sequence generated will test the system conditions to determine the action to be taken. In the diagnostic mode, error conditions which are hardware detectable are visible to microdiagnostics. The microdiagnostics control

the action to be taken. A software error signal on the other hand causes the control store to branch to address binary 1. This address is the start of the software error reporting sequence which is under micro-program control.

Referring once again to Figure 13c the E field 1326 is a 3 bit field for the branch code as previously described. The branching and/or micro-operation field 1327 is comprised of the A, B, C, D, and L fields (also shown on block 1335 of Figure 13b) wherein the A field is the upper 6 bits of the next address, the B field is the middle 4 bits of next address of the mask field 64 tests, the D field is another 6 bit test field for 1 of 64 tests, and the L field is the least significant bit. The K field 1328 is a 14 bit field of which 6 bits are for the constant field, 4 bits are for a constant or steering field, and 4 bits are a steering field for a constant. The data to bus field 1329 is comprised of the QA field having 4 bits for controlling information to the QA portion of the QMB bus 1344 and the QB field has 4 bits for controlling information to the QB portion of the QMB bus 1344. The F field 1330 is a 32 bit field which is coded to generate micro-operation subcommands. The P field 1331 is comprised of 4 bits reserved for checking.

In operation the micro-instruction words are stored in the control store array 1333. During a cycle of operation, the control store array is addressed by the contents of the KS address register 1337. This causes the contents of the location specified by the address to be read into the group of read latches 1357. Portions of the word contents of the read latches are distributed or transferred to storage registers within each of the functional units in the CPU. Each functional unit includes decoding logic circuits for generating the requisite subcommands specified by the control store word under control of the system clock source. In general decoding is performed within each functional unit in the CPU rather than being performed centrally in order to minimize the decoding time and to reduce the number of cables which would be normally required for transmitting command signals if decoding were performed centrally. Additionally, the decoding is done within each unit to avoid timing problems

arising from differences in cable delays. Further, by decoding subcommands with each unit, those signals which are representative of certain conditions existing within the functional unit are required for the generation of certain subcommand signals do not have to be returned to the CIA unit 1302. A typical decoder unit 1359 is shown in Figure 13b as receiving various fields from micro-instruction words and generating micro-operation signals a, b, c, d, . . . q, r. A typical micro-operation decoder 1359 receives commands from a micro-instruction word. The field from the micro-instruction word is decoded and sets one of a plurality of lines s, t, u, . . . y, z high. A matrix is formed by having predetermined control line impedance coupled to the s-z lines at points α, β, γ . . . Ψ, ω. Typically then when the field from a micro-instruction is decoded one of the lines s-z goes high. Since the black dots shown in the matrix by Greek letters α through ω represent impedance coupling between the two sets of lines, any electrical signal propagating along any horizontal wire will be coupled through to propagate along the vertical wire a-r where an impedance coupling (black dot) is indicated. Each vertical line a-r then may be coupled as one input to one each of AND gates 1360-1365. Other input signals may also be coupled to AND gates 1360-1365. Other input signals may also be coupled to AND gates 1360-1365 including a timing signal t_s from the central timing unit. Accordingly as each timing signal t_s goes high those gates which have all the other input signals high will be enabled and will provide micro-instruction signals to predetermined functional units in the CPU. For example, if a command 1341 from read latch 1357 is decoded and a horizontal line is high the a, b, c, and q vertical control lines will be high and AND gates 1360, 1361, 1362, and 1364, will be enabled as the t_s timing signal is applied to these gates sequentially. Hence the combination in which the vertical control lines are coupled to the horizontal control line at different points represented by the Greek letters α through ω rep-

resent a permanent switching matrix for supplying micro-operation signals to the central processing unit CPU for controlling the functional units within the central processing unit by micro-instructions furnished from the control store array 1333. Thus permanent firmware having an alterability feature can be built into the machine of this invention by merely specifying the sequence of micro-operations that is required as a capability of the computer system.

Under normal conditions data is written into the control store array 1333 via the CPU write-data register also known as the local register YO 1343. A control flop (not shown) defines whether the uper half or lower half of the storage array is to be written. Data from the control and load unit CLU 1304 arrives at the CIA/CSU via the maintenance bus QMB 1344 and is buffered by the storage local register YO 1343 before being written into the control store array 1333. The storage local register 1343 is time shared as both a read and a write local register. The multiplexor KQM 1345 may be controlled by either the maintenance panel 1355 or by micro-diagnostics and provides a read out path from the registers connected to it. A compare register KP 1350 is provided for non-functional use and is used mainly for maintenance purposes and is utilized along with compare logic 1352 and decode logic 1351.

DISPATCHER FIRMWARE FOR CONTROL OF PROCESSES

The dispatcher is a firmware/hardware unit whose main purpose is to manage the various queues of processes, and to switch between processes and includes the updating of the queues of processes, the process control blocks PCB's, the running process word in the system base and the register of a new process. It also delivers messages to a process which is actually waiting for it on a semaphore (after a V-operation, a simulated V-operation for the IOC or for the exception handler). It also enqueues a message on a semaphore after a P-

operation which freed a message link when a process is waiting on the free-link semaphore to deliver its message.

The dispatcher unit further calls on the native mode instructions firmware after the "rolling-in" of a process executing in native mode, or after a "contest" if the current process remains running and is executing in native mode. It also calls on decor extension firmware for:

(a) a temporary call during the rolling-out of a process executing in that decor extension;

(b) a temporary call during the rolling-in of a process executing in that decor extension;

(c) a definitive call at the end of the rolling-in of a process executing in that decor extension; and,

(d) a definitive call after a contest if the current process remains running and is executing that decor extension.

Moreover the dispatcher places the system in the idle loop when there is no running process.

There are several ways of entering or leaving the dispatcher as follows:

(1) The initialization procedure (SIP) provides entry as the last step, (See Patent Application No. ———— entitled System Initialization Procedure, assigned to the same assignee named herein.)

(2) The Start and Suspend Instruction provides for the entry into the dispatcher. The Start instruction starts a process and the Suspend instruction terminates a process. (See Patent Application No. ———— entitled Start and Suspend Instruction.)

(3) The P & V operations provide for entry into the dispatcher. The P op picks up a message from a semaphore and if no message then the process goes into wait state. (See Patent Application No. ———— entitled "P and V Operations of Semaphores".)

Briefly then, the dispatcher is the main mechanism which manages the processes and accordingly the process control blocks PCB's by deciding which process is to be run and then taking the appropriate action, such as rolling-out the currently running process (i.e. write into the PCB all information relative to the currently running process contained in hardware registers, scratch pad memory, etc.), and rolling-in the new process (i.e. write

from the PCB to various hardware registers, scratch pad memory, etc., all information necessary to run the new process.)

The representation of the functions performed by the dispatcher are shown in flow-charts Figs. 14a-14i. For example, block 1402 of Fig. 14a is a representation of the function performed by the dispatcher wherein a micro-program word is delivered by the control store unit, and upon decoding by decoder 1359 controls the applicable portions of the CPU via an appropriate series of micro-operation signals 1360, 1361, etc. to retrieve the IPQW from the system base in storage subsystem 102, and transfer it to scratch pad memory 1315. Simultaneously the dispatcher fetches 1404 the GO segment descriptor (see Figure 12) from the G table of segment descriptors pointed to by the G table word GTW in the system base. Bits 16-31 of the IPQW word contain a 16-bit positive integer which is the displacement from the base of the G segment number 0, referred to as the GO segment, to the head (first byte) of the Q/PR/RDY, queue of processes ready. If bits 16-31 of the IPQW word is 0, 1403 the ready queue is considered to be empty. If the ready queue is empty it indicates that there is no process currently waiting in the Q/PR/RDY and the ready-queue is empty. The next question to be decided in decision block 1405 is whether or not a process is currently running in the machine by determining whether or not the Vacant Indicator is set. (The Vacant Indicator is a flip-flop located in auxiliary memory 1317a; it is set when there is no current process CJP running in the processor). If the Vacant Indicator is set (i.e. there is no process currently running) and since it was previously determined that there is no process in the ready-queue waiting to use the processor then the machine goes in the idle state 1406. However if there is a process currently running in the machine, but there is no one waiting to use the machine, the current process accesses its next instruction 1407.

Returning now to decision block 1403, of the flow chart of Fig. 14a if there is a positive integer in the pointer area of the IPQW (i.e. bits 16-31) the head of the ready

queue pointed to by the IPQW word in the GO segment is fetched to the scratch pad memory. (Note: To avoid repetition and in the interest of clarity the intermediate functions of the dispatcher in conjunction with the control unit and CPU will be omitted; however it is to be understood that such intermediate functions as previously described by example are present typically.) Up to this point it has been determined that there is some process waiting in the ready-queue. Before further action can be taken it is necessary to determine if there is a process currently running in the central processor. This is determined at decision block 1410 of the flow chart representation if there is no process currently running in the central processor (i.e. no CJP) the head of the ready queue is run 1412. However if there is a process running in the central processor the dispatcher must determine who has priority—the currently running process or the head of the ready queue? Accordingly therefore the priority byte of the current process (CJP) which is located in the running process word of the system base 600 or in process main word 0 PMW 0 of PCB 400, is fetched 1413. A decision is then made 1414 as to whether or not the current running process CJP is of lower priority than the new process NJP waiting at the head of the ready queue? (See decision block 1414.) If CJP is not of lower priority than NJP, CJP remains in control of the central processor and the contest indicator is reset 1415. (The contest indicator is always set to 0 except when one or more new processes have been placed on the ready-queue since the beginning of the last instruction executed on behalf of CJP thereby creating the possibility of a conflict; under these conditions the contest indicator is set to binary 1). Before the current process CJP is permitted to continue however, and execute further instructions a determination is made whether or not the CJP is running in decor extension mode 1415. If CJP is running in decor extension mode then the next instruction is executed in emulation mode (i.e. decor extension) and if it is not running in decor extension mode then the next instruction is executed in native mode. Returning once again to decision block 1414, if the NJP at

the head of the ready queue has a higher priority than the CJP (i.e. its priority number is lower than CJP's priority number, the current running process CJP is "rolled-out" of the machine and the new process NJP is "rolled-in" to the machine. Accordingly a firmware priority subroutine PRIQ 1418 directs the enqueueing of the current process CJP into the ready-queue by LIFO priority and by priority number by first "rolling-out" the CJP under the direction of the firmware subroutine RLLO 1419. The RLLO subroutine directs the writing of the information of the CJP that is stored in the general registers, base registers, scientific registers, T-register, status registers, and instruction counter, back into the appropriate storage areas of the process control block PCB in main memory, and it directs the updating of the RUA. Moreover the DEXT number of process main word 0 (PMW0) in PCB 400 is updated 1420.

The new process NJP is now ready to be "rolled-in". The boundary address register BAR is fetched 1422 and the running process word RPW is fetched from address BAR plus 56 of the system base. See block 1423. The name of the new process NJP is next written into the running process word RPW and since the name of the new process NJP was written in the process link PL of the Q/PR/RDY, the name in the process link PL is therefore now placed in the RPW, block 1424. Therefore NJP from the ready-queue now becomes CJP and is entitled to control the central processor and accordingly is no longer waiting in the Q/PR/RDY and must be dequeued by taking its name out of the process link PL of the Q/PR/RDY, block 1425. When this is done the queue of processes ready Q/PR/RDY is updated by the firmware subroutine UQLK, 1425a. Accordingly the JP number of the process which was just taken out of the machine is placed in a process link in the Q/PR/RDY, since it now no longer has control of the machine and must wait for it, 1426. At this point the transformation of giving this control of the central processor to the new process and placing the old process in a ready queue is accomplished and because there is a process (the new CJP) in control of the central processor, the Vacant

Indicator is set to 0, 1427; if on the other hand there were no CJP in control of the central processor the Vacant Indicator would be set to 1. At this point the assignment of the processor is complete and a new process has acquired the central processor whereas the old process was placed in a ready queue; however, the new process is not ready to run yet for the hardware of the central processing unit, 1306 of Figure 13a, such as the general registers 1307, the base registers 1308, scientific registers 1309, the T-register 1310, the status registers 1311, and the instruction counter 1312, must be supplied with control information from the process control block of the new process.

Accordingly the firmware subroutine 1430 first fetches PMW 3 from the PCB (Figure 4) to scratch pad memory 1315, and then fetches PMW 0. The MBZ field of PMW 0 is checked 1433 and if it is not binary 0 an illegal PCB exception results. However if the MBZ field of PMW 0 is zero then PMW 1 is fetched 1434. Again the MBZ field of PMW 1 is tested to determine whether or not it is a binary 0. If it is not binary 0 there is an illegal PCB exception; whereas if it is equal to 0 the dispatcher proceeds to C.

Accordingly address space word 0 ASW 0 is fetched from the appropriate space in the PCB and the segment table word sixe STWSZ is tested 1437 to determine whether or not it is less than 7. If it is greater than 7 an illegal PCB results if it is less than or equal to 7, then ASW 1 is fetched from the PCB, block 1438, and its STWSZ field is tested 1439 to determine whether or not it is less than or equal to 8. If its field is greater than 8 then an illegal PCB results; however, if its STWSZ field is equal or less than 8 the exception word EXW is fetched 1440, and its MBZ field is tested to determine whether or not it is equal to 0. If its MBZ field is not equal to 0 an illegal PCB results whereas if it is equal to 0 the stack word SKW is fetched 1442, and its MBZ field is tested 1443 to determine whether or not it is equal to 0. If the MBZ field is not equal to 0 then an illegal PCB results; whereas if it is equal to 0 the instruction counter word ICW is fetched from the PCB and placed

in the instruction counter IC and its TAG field is tested to determine whether or not it is equal to 0, 1445. If the TAG field is not equal to 0 then an illegal PCB results. However if the TAG field is equal to 0 the MBZ word is fetched 1446 and its MBZ field (bits 0-31) is tested to determine whether or not it is equal to 0, 1447. If it is not equal to 0 then an illegal PCB results; whereas if it is equal to 0 stack base words 0, 1 and 2 SBW 0, 1 and 2 are fetched 1448. The contents of the 8 base registers in the base register saving area of the PCB is then fetched 1449 and stored in the machine's base registers 1308. Then the contents of the 16 general registers from the general register saving area of the PCB are fetched 1450 and stored in the machine's general registers 1307. Before fetching the contents of the scientific registers however a check is made of the capability byte of process main word 0 (PMW 0) to determine whether or not scientific mode is being utilized 1451. If scientific mode is being utilized then the contents of the scientific registers from the scientific register saving area of the PCB are fetched and stored 1452. The firmware then proceeds to make a check of the capability byte of PMW 0 to determine whether or not the accounting mode is being utilized 1453. If the accounting mode is being utilized (i.e. accounting bit of capability byte set to binary 1) the accounting words exists in the PCB and the ready time account word RTA is updated. Then the firmware proceeds to determine whether or not the DEXT number is set to 0, 1454. If it is not set to 0 it indicates that the machine may be in emulation mode (i.e. the decor extension capability is being utilized) and accordingly the DEXT number of PMW 0 is checked 1455 to determine whether or not it is greater or less than the DETSZ field of process main word 3 and if it is greater than the DETSZ field an illegal PCB exception 1456 results since the DEXT number is less than the DETSZ field but not equal to zero the machine is performing in legal emulation mode and proceeds to F. Returning to decision block 1454 if the DEXT field is binary 0 then native mode is being performed and the machine fetches STW's 1457. The residual time out word RTO word of the PCB is

fetched 1458 and the process timer is loaded, with the time that the CJP may spend in the running state.

Up to this point either (a) a new process NJP has been "rolled-in" to take control of the CPU when there was an old process CJP in the machine and the new process NJP was at a higher priority than the old process CJP or (b) there was no CJP in control of the CPU and the head of the ready queue was rung. Briefly under condition (a) the CJP was taken out of the RPW and placed in a process link PL in the Q/PR/RDY, and the NJP in a process link PL in Q/PR/RDY was placed in the RPW thus effectively switching the positions of the 2 processes giving control to the NJP which now becomes the CJP and taking control away from the old CJP. Then the PCB of the NJP was accessed and the information required to run NJP (now the CJP) was placed in scratch pad memory or the array of registers in the ACU.

If there was no CJP in control of the CPU (condition b), then the head of the ready queue was run—i.e. the NJP became the CJP because of the dispatcher taking the NJP from the process link PL from the head of the ready queue and placing it in the RPW word. By doing this a process link PL was left empty in the Q/PR/RDY and it is required to take it out.

Accordingly beginning now at decision block 1461 the firmware determines whether or not there was a CJP in control of the CPU and if there was a free process link (FPLS) it was accessed and enqueued and the CJP was written therein. However if there was no CJP in control of the CPU state byte of PMW 0 of the NJP is updated 1460 and again there is a determination whether or not there was a CJP in the machine 1463. If there was no CJP in control of the processor the process link of the NJP (which was in the Q/PR/RDY and is now in control of the machine) is taken out of the Q/PR/RDY 1466 (i.e. dequeued from the Q/PR/RDY) and becomes a free link semaphore FLSP and is now enqueued in the free process link queue (805 on Figure 9) and becomes a part of the free process link queue 1466a. The contents of the boundary address register BAR are fetched 1464 and the

running process word RPW of the NJP (now the CJP) located at BAR plus 56 of the system base is updated by placing the NJP identification in the RPW, block 1465. The Vacant Indicator is set to 0 in the case where there had been no CJP. Next the Contest Indicator (i.e. a flip-flop in auxiliary memory 1317a used to indicate a possible conflict in priority between the CJP and a process placed in the ready queue) is set to 0, 1467 and the Segment Associator (AS 132 on Figure 1) which is a typical content addressable memory is cleared, 1471; then the process mode is entered 1470. (A process mode indicates that exceptions are handled by the process which is operating in the processor rather than by the operating system). The firmware then continues to CAB 1480 and the asynchronous trap bit AB is checked to determine whether or not it is set to binary 1, 1481. If the AB bit is set to binary 1 then a check is made 1482 to determine whether or not the process ring number PRN is greater than or equal to the asynchronous trap bit ARN. (The AB and the ARN are located in the priority byte of the PCB of each process and is meaningful when the process is in the running state; the AB and ARN are obtained from the RPW located at BAR plus 56 of the system base). The AB and ARN in BAR plus 56 of the RPW are reset since the next step 1484 is to proceed to an asynchronous trap routine which would take care of the conditions that cause the asynchronous trap bit or the asynchronous ring number to be set in the first place, and if these were not reset then at the next pass by the firmware it would give an indication that something was wrong where actually there was nothing wrong, and hence would always proceed to the asynchronous trap routine 1484 and would never execute. Going back now to decision blocks 1481 and 1482 if the AB bit is not set or AB bit is set and PRN is not greater than ARN, then the firmware would proceed to determine in what mode the processor would run—in the normal mode or in the emulation mode. Accordingly the DEXT number is checked to determine whether or not it is set to 0 and if it is set to 0 the normal mode of the machine is run 1487. However if the DEXT number is not set to 0 the emulation mode is run 1486.

The Switch System Base Instruction

Referring now to Figure 15a there is shown the structure of the switch system base instruction. The operation code OP 1501 is similar to the operation code of any instruction and indicates the operation to be performed. The complementary code C 1502 is used in conjunction with the op-code 1501 when the op-code shares more than one instruction 1501 and the complementary code 1502 determines the precise instruction to be performed. For example, the P and V operations have 12 instructions and they will use one op-code. The complementary code together with the op-code determine which of the 12 P and V instructions is to be performed. The address syllable AS 1503 of the switch base instruction is the address pointing to a new copy of the system base which is to be loaded into main memory and also scratchpad memory. Accordingly, this address points to an entirely new system base. Typically a 60 byte zone, starting at the address specified by the address field of the switch system base instruction, is transferred into the system base area which starts at address BAR. This transfer is also made into the scratchpad memory. This zone, therefore, is the new system base which replaces the previous system base. The running process, therefore, disappears because its PCB is not updated and no link manipulation is performed. The processor is then placed in the VACANT state, and the dispatcher entered.

It is the system software responsibility to prepare the new information for the system base and the new GO segment before executing the SWSB instruction. (See applications included by reference herein numbers 12-16). For that reason, the SWSB instruction must be entered with the hardware gate HG locked, otherwise it returns a condition code CC without taking further action. Upon completion of SWSB instruction the HG will be left unlocked. Also upon completion of the SWSB instruction the vacant switch is set to 1. This indicates no process is running and as a result the system will go to the head of the ready queue for the internal process queue word IPQW and pick out the pointer to the queue of processes ready. It will then proceed to the head of

the ready queue and start up a new process. The process which executed the SWSB instruction will terminate because a vacant condition has been placed in the vacant indicator, which means that there is no running process.

Referring now to Figures 15b and 15c the SWSB instruction in firmware will be described in greater detail. The SWSB instruction starts 1504 and a determination is made 1505 whether or not the process ring number PRN is 0. This is necessary so that the SWSB instruction executes with the highest privilege. If the PRN is not 0 then a privileged instruction exception 1506 occurs. If on the other hand PRN is equal to zero the hardware gate located in auxiliary memory 1317a of the ALU 1317, is checked 1507 to determine whether or not it is locked. It is important that the HG is locked during the execution of the SWSB instruction. When the HG is locked no I/O interrupts or exceptions are permitted. If such exception or an error condition occurs while the HG is locked the system will crash. The system crash is a more desirable alternative while the system base is being changed, then permitting the interrupt with the system base only half changed. Accordingly, if the HG is not locked the condition code CC in one of the status registers is set to 2, (block 1508) which in this case is the code indicating the hardware gate HG is not locked.

Accordingly, the SWSB instruction is not allowed to proceed 1509, until the hardware gate HG is locked. If on the other hand the hardware gate HG is locked a determination is made as to whether or not the new system base word (i.e. address syllable AS) is quad-word aligned. If the AS is not quad-word aligned then the condition code CC is set to 3 indicating this condition and the SWSB instruction terminates 1509. On the other hand if the AS is quad-word aligned the condition code CC 1512 is set to 0 which indicates that the SWSB instruction was successful and can proceed to perform the move operations 1513 from the new system base image into the system base area located at BAR. Finally after the new system base image has been transferred into the BAR system base the hardware gate is unlocked 1514,

the vacant indicator in the auxiliary memory 1317a is set to 1, (block 1515) indicating no running process, and finally the instruction is complete 1516.

Figure 15c is a more detailed flow diagram of block 1513 which is the step that transfers the new system base image to the BAR system base area, and includes loading of the scratchpad memory S.P.M. with information from new system base image.

Referring to Figure 15c the start 1517 of SWSB instruction is actually at block 1513 of 15b. Typically 60 bytes of the new system base image located at address AS are transferred 1518 to the system base whose address is indicated by the boundary address register BAR. A new G table word GTW from the new system base image is loaded 1519 into scratchpad memory S.P.M. in auxiliary memory 1317a. Then a new J table word JTW is loaded 1520 in S.P.M. Finally a new internal processor queue word IPQW is loaded 1521 into S.P.M., and that portion of the instruction completes 1516 and continues to block 1514, where the operations are as previously described.

GLOSSARY OF TERMS

Absolute Address —the physical address of a hardware byte in main memory store.

Address Development —a hardware function which operates on a number of address elements to compute an absolute address which is used to refer to a location in main storage.

Addressing —locating an object by any of a number of virtual, logical, and physical means.

Address Space —the set of logical, segmented addresses corresponding to a process that the CPU is permitted to transform into absolute addresses during execution of that process.

Address Space Word —one of two words in a process control block which point to the segment table word array. The segment table word array defines the segment tables associated with the process.

Address Syllable —a logical address recognized by CPU hardware, normally an operand of an instruction.

Analyzer/Translator —that part of the static linker that first passes the control language stream to the linker, then, if the control language is error-free, the analyzer/translator converts it to tables and other structures for the static linker to use.

Asynchronous Processing —concurrent activation of more than one process.

Auxiliary Memory —is comprised of 64 flip-flops for storing miscellaneous states of the system and is located in ALU.

Base Register —the primary element in segment addressing, referenced by number in every address syllable.

Boundary Address Register —a visible hardware register that defines the lowest memory address accessible to firmware/software.

Call —see procedure call—Inward Call-Outward Call.

Central Process Unit —a part of a computer that includes circuits controlling the interpretation and execution of instructions.

Channel —a means of communicating between the processor subsystem and a peripheral subsystem. There are two types of channels, physical and logical. A physical channel is the hardware connection between the input/output control unit IOC and the peripheral control

unit PCU. A logical channel is a software communications path between main memory and a single peripheral device.

Channel Command Entry —an instruction in a channel program.

Channel Command Word —an element of a channel command entry. Two channel command words constitute a channel command entry.

Channel Program —a sequence of instructions that cause a specific input/output i/o operation to be performed by a peripheral device.

Complementary Code —a code in some instructions which together with the operation code defines the function of the instruction.

Compilation Unit —the single object-code module that results from compiling or assembling one procedure in any high level language processor. The CU is not executable unit it is linked.

Concurrency —apparent simultaneity.

Condition Field —a 4-bit field in a branch instruction. Each bit of the field corresponds to a setting of the condition code in the status register bits 0 and 1.

Contest Indicator —located in auxiliary memory and indicates a new entry was placed in the ready queue.

Data Address Space —the set of logical record addresses accessible by a process through data management; consists of four elements; logical fields, logical records, data base pages, and files.

Data Descriptor —an information structure which is used in indirect addressing and as a description in data manipulation.

Decor —the functional characteristics or architecture of a computer system.

Decor Extension Mode—a mode of operation permitting the operating of the system in emulation mode.

Descriptor —a 32- or 64-bit field used in developing the address of any data field. It is one of many pointers in the addressing scheme of the invention.

Descriptor Block —a data block containing any number and variety of descriptor declarations except semaphore descriptors. No other data is allowed.

Device Adapter —electronics which adapts a specific device to a peripheral controller.

Device Adapter Interface —the interface between a device adapter and its associated peripheral controller(s).

Device Control —a software facility which handles device specific characteristics.

Dispatcher —the firmware responsible for dispatching processes in the CPU.

Dispatching —a field within the instruction format or a data descriptor which provides relative position information within a segment.

Displacement Addressing —a form of direct addressing in which the operand defines a location relative either to the base of the immediate segment or the current ation counter.

Dynamic Addressing —addressing which results in resolution, during execution, of references defined within the process group.

Editor —that part of the static linker that format and outputs any or all of the information that results from linking a particular procedure.

Effective Address —a logical, as opposed to physical, address consisting of a segment table number, a segment table entry, and a segment relative address. These elements show the way to a particular segment descriptor, which eventually points to the segment being referenced.

Emulation Mode —operation of a computer so as to initiate another computer, and appear to operate in the other computer's environment.

Exception —an exception occurs when the hardware detects the existence of some condition that has been specified as requiring special handling.

Event —anything observed in a system by a process which may be of interest to some other process.

Firmware —that part of a hardware unit implementation which utilizes a form of micro-programmed control.

G, D Name —the name used by the hardware to refer to a semaphore. G, the G-segment name is the number of the entry in the G-table containing the segment descriptor, D is the relative address in the segment.

G-Segment —a segment containing semaphores which may be addressed using the G, D name form and associated mapping facility (the G-table).

G-Table —a table used for storage mapping (locating a G-segment) when the G, D name form is used.

Gating —a facility for controlling entry to procedures in a (gated) segment through the use of procedure descriptors.

General Register —a 32-bit register available to executing processes which generally holds binary or bit string data. Certain general registers can be used for indexing (GR8 through GR15).

Index —a data structure which is maintained and operated upon by the system. The structure is visible to the user only in terms of a key which he presents for the purpose of making available or positioning to a specific record occurrence.

Indexing —modification of an address by a signed arithmetic value by adding something to something.

Index Register —a general register used for indexing (GR8 through GR15).

Indirection —a method of addressing whereby an address is used to obtain another address, rather than the actual data.

Indirection to Base —the form of indirection whereby a data descriptor containing a base register number and a displacement is found at the address referenced.

Indirection to Segment —the form of indirection whereby a data descriptor containing a segment address is found at the address referenced.

Indirect Segment Descriptor —a type of segment descriptor which contains information for addressing another segment descriptor, rather than a segment.

Instruction —the unit of execution of a central processor visible to a programmer.

Instruction Counter	—a register associated with an executing process which contains the segmented address of the next instruction of the procedure being executed.
Interleave	—sequential accessing of memory modules to reduce memory access time.
Interrupt	—the discontinuance of a process in the system due to an event occurrence.
Interrupt Process	—a process which is activated in response to an interrupt.
Inward Call	—when one procedure execution in one ring number calls another procedure to execute in a lower ring number.
I/O Controller	—a unit which provides the basic control for a specific I/O subsystem.
I/O Processor	—potentially asynchronous system processes which deal primarily with the movement of data between peripheral storage or I/O devices and main storage.
J.P. Tables	—a collection of logical addresses for locating a process control block.
Job	—a unit of work for the system.
Job Control Language	—a language for describing a job and its component parts.
Job Step	—the major subdivision of a job leading to the execution of a program defined by a JCL statement.
Job Step Table	—a system table used to map the first part of a process name, i.e., to locate the process group table.
Linked Module	—the output of the static linker. It is a consolidated group of compilation units (CU) whose cross-references were resolved by the static linker.

Linking	—see static linking.
Logical Channel	—see channel.
Logical Processor	—the collection of hardware resources and control information necessary for the execution of a process.
Main Storage	—all addressable storage from which instructions can be executed or from which data can be loaded directly into registers.
Magnetic Tape Controller	—the elements of the peripheral subsystem related to magnetic tape devices.
Mass Storage Controller	—the elements of the peripheral subsystem related to mass storage devices.
Memory Management	—operating system facilities for the allocation, relocation, and disallocation of physical memory.
Message	—information transmitted from a source or to a destination—neither being a file.
Microinstruction	—same as microinstruction word, and control store word.
Micro-program	—a set of machine codes used to implement the control functions of a processor.
Multiplex	—to share a hard resource, such as memory, usually by time division.
Multi-programming	—the concurrent execution of two or more programs by a single computer.
Native Mode	—operation of a computer in its native environment.
Offset	—in address development, the number of bytes after the beginning of the segment at which an addressed part of the segment starts.

Operating System	—a system of basic software designed to support the efficient operation of user software.
Outward Call	—when one procedure executing in one ring number calls another procedure to execute in a higher ring number.
Peripheral Control Unit	—a stand-alone micro-programmed processor which executes channel programs to perform I/O operations.
Peripheral Subsystem Interface	—a standard interface used for transfer and control between freestanding peripheral control units and I/O control units.
Physical Channel	—see channel.
Physical Data Structure	—a complete definition of the organization of data as it is physically recorded on media.
Physical I/O	—that part of the operating system which initiates and controls the transfer of data between memory and peripheral or terminal devices.
Physical Storage	—hardware used for the storage of data. It is composed of various types of recording media and the recording/reading hardware.
P-operation	—a machine instruction involving a semaphore which causes a process to enter the wait state or to receive a message.
Privileged Instruction	—an instruction which must be executed in ring 0.
Procedure	—a named software function or algorithm which is executable by a processor without concurrency: a Fortran subroutine, a Cobol program, a PL/1 internal procedure.

Procedure Call	—a mechanism for creating a stack frame within a stack segment.
Procedure Descriptor	—a word containing the location and entry point of a procedure.
Procedure Segment	—the type of segment whose contents represents a procedure, procedures, or part of a procedure.
Process	—the ordered execution of instructions without concurrency by a processor, either central or i/o.
Process Address Space	—the memory addresses which may be referenced or to which control may be transferred during a particular process.
Process Control Block	—a data structure (hardware defined and recognizable) that contains information needed to specify the state of a process at any given time.
Process Group	—a related set of processes, usually those necessary for the performance of a single job-step.
Process Group Builder	—that part of the static linker whose actions yield the linked module.
Process Group Loader	—an operating system facility that makes final resolutions in a linked module to create an executable process group then it loads the process group into memory and starts its execution.
Process Group Table	—see P Table.
Process Link	—an entry in a ready process or waiting process queue.
Process State	—the dynamic condition of a CPU process e.g., running, ready, waiting, or suspended.
Process Switch	—the hardware function that disassociates one process from the central processor and associates another.

Process Synchronization —the functions that act to coordinate the asynchronous activity among processes. These functions generally involve semaphores and the P and V operations.

Processor —a unit capable of receiving data, manipulating it, supplying results, and controlling its sequence of operations in response to stored programs: short for central processor, generic for central processor, peripheral processor, or hardware/software processor.

Program —the specification of the procedural and associated information necessary to solve a problem, i.e. the ordered collection of instructions which is executed by the computer and operates on data to perform a user's job or some phase of that job.

Program Execution —the activity of a process in conformance to program specification.

Protection —the function, enforced by hardware and software, of preventing processes from interferring with each other or sharing each other's address space in an unauthorized manner.

P-Table —a hardware defined data structure containing entries, each pointing to the origin of a process control block. The P-table entries point to the total set of PCB's comprising a process group at any given time. Also called process group table.

Q/PR/RDY —queue of processes ready i.e. a ready-queue.

Queue —an ordered list of entities awaiting information, process or availability.

Ready State	—a process state in which no processor is associated but all necessary resources, except a processor, are available for changing to the running state.
Relative Address	—the incremental/decremental location of some object relative to another.
Relocation	—movement of a segment from one location in main storage to another including the adjustment of all required references to its previous location.
Return	—the function and operations necessary to resume the action of a procedure at a point immediately following the point where it entered some other procedure.
Ring	—a protection attribute of a segment that restricts the read, write, and execute access of process to that segment. A level of privilege of a process for read, write or execute.
Rolling-in	—performing the operation necessary to give control of a processor to a new process.
Rolling-out	—performing the operations necessary to take away control of a processor by a process.
Running State	—the dynamic condition of a process (connected to a CPU) that is executing.
Scratch Pad Memory	—256 locations in ALU for storing CPU control information-also known as a local store unit.
Scheduling	—determining the sequence of operations.
Scientific Registers	—a 64-bit register used in the manipulation of binary floating point numbers. There are four scientific registers in the invention when the system has the scientific option.

Segment	—a contiguous main memory space treated as a unit.
Segmentation	—the use of a division of main memory into logical groupings called segments rather than into a single linear store.
Segment Base	—the current origin of the segment. This is one of the fields in a segment descriptor word.
Segment Descriptor	—entry in a segment table defining the characteristics of a segment or pointing to a segment descriptor.
Segmented Address	—see effective address.
Segment Number	—the identification of a particular segment consisting of a segment table number (selecting one of the segment tables of the process) and a segment table entry of that chosen table.
Segment Relative Address	—during address development, the final value to be added to the segment base to obtain the absolute address.
Segment Table	—a table containing segment descriptors used to establish the addressability of a process to its segments. The collection of segment tables of a process defines the address space for that process.
Segment Table Entry	—an item in a segment table. Each segment table entry is a segment descriptor.
Segment Table Word	—an entry in a segment table word array pointing to the segment tables of a process. The array is pointed to by one of two address space words in the process control block.
Semaphore	—data structures used to control communication between processes.
Semaphore Block	—a data block containing only semaphore descriptor declarations.

Stack —a mechanism that accepts stores, and allows retrieval of data on a last-in-first-out basis. Comprised of a number of contiguous portions called stack frames.

Stack Active Area —the portion of the current stack frame that holds a pointer to the saving area, the caller's instruction counter, parameter space and local variables—in short, the data of the current frame most likely to be useful to execution of the called procedure.

Stack Base Words —a group of three words in a process control block which contain the segmented address of the stack segments for rings 0, 1, and 2 associated with the process.

Stack Frame —a contiguous part of the stack segment holding the data stored for a single procedure call.

Stack Operations —the push-down and pop-up operations of storing and retrieving information to/from the stack. In practice, manipulation of the pointers used to reference the information stored in the stack.

Stack Overflow —the action of attempting to store more information into the stack then the available space can hold. This is detected by hardware.

Stack Register —see T register.

Stack Segment —the segment of a process that contains its stack.

Static Linking —an intermediate step between compiling a source procedure and executing the object form of that procedure; linking resolves the external references made to and from that procedure.

Status Register —an 8-bit register which specifies the status of the procedure currently being executed.

Swapping —releasing the main memory space occupied by a segment by writing the segment out to secondary storage. The actual write to secondary storage may be suppressed if the segment has not been modified since the last time it was read in.

Synchronization —having a fixed relationship in time; synchronized execution of two or more processes.

System Base —a fixed area in main memory which acts as a root for all information structures.

Task —the smallest unit of user-defined work consisting only of a single stream of non-concurrent instructions.

T Register —a software visible hardware register which contains the segmented address of the top of the procedure stack.

Unit Record Controller —the elements of the peripheral subsystem related to unit record devices such as card equipment, paper tape equipment, and line printers.

User Process Group —the internal representation of any job step, as opposed to a system process group which exists independently of any job.

Vacant Indicator —a hardware structure that indicates that the process which had control of the CP has suspended itself i.e. no CJP in the CPU. Located in Auxiliary Register, of the ALU.

Virtual Memory —an addressing concept that allows a programmer to code without regard to physical memory size; virtual memory manager automatically manages the exchange of segments between main memory and secondary storage.

V-Operation —a machine instruction intended for interprocess signaling the completion of an event. A V-OP operates on a designated semaphore.

ABBREVIATIONS

ACU —address control unit
ALU —arithmetic and logic unit
AS —address syllable
ASW —address space word
BAR —boundary address register
BCD —binary coded decimal
BR —base register
CCE —channel command entry
CCU —channel control unit
CMRN —the caller maximum ring number
CCW —channel command word
CET —current state entry time
CIA —control store interface adapter
CJP —currently executing process
CPU —central process unit
CU —compilation unit
CSU —control store unit
D —displacement
DMU —data management unit
DA —device adapter
EAR —effective address ring
EXW —exception word
GR —general register
GTW —G-table word
IC —instruction counter
IFU —instruction fetch unit
I/O —input/output

IOC —input/output controller
IPQW —internal process queue word
IR —index register
ITBB —indirect to base register
ITS —indirect to segment
J —job
JCL —job control language
JTW —J-table word
LCT —logical channel table
LIFO —last-in-first-out
LSU —local store memory or scratch pad memory
MBZ —must be zero
MOS —metal oxide semiconductor
MAXR —the maximum ring number at which a procedure may execute MAXR is found in the segment descriptor of SEGEP.
MSC —mass storage controller
MTC —magnetic tape controller
NFS —non-functional status
NPRN —new process ring number
NJP —new process replacing the currently executing process
PCB —process control block
PCT —physical channel table
PCU —peripheral control unit
PL/I —programming language/one
PMW —process main word
Q/PR/RDY —queue of processes ready
RD —read ring
RHU —reserved for hardware use
RPW —running process word
RSU —reserved for software use
RTA —ready time accounting
RTO —residual time out
SBW —stack base word
SEG —segment number
SKW —stack word
SEGEP —the segment which contains the entry point as found in the procedure descriptor
SEGPD —the segment which contains the procedure descriptor

SR	—scientific register
SRA	—segment relative address
STE	—segment table entry
STN	—segment table number
STR	—status register
STW	—segment table word
STWA	—segment table word array
T	—T-register (stack register)
URC	—unit record controller
WR	—write ring
WTA	—waiting time accounting

Having shown and described one embodiment of the invention, those skilled in the art will realize that many variations and modifications can be made to produce the described invention and still be within the spirit and scope of the claimed invention.

What is claimed is:

* * *

For Claims 3, 5 and 6, see Rejected Claims 3, 5, and 6.

DECLARATION AND POWER OF ATTORNEY — (for use with sole or joint patent applications accompanying or not accompanying the application papers)

As a below named inventor, I hereby declare that the information given below is true. That I believe that I am the original first and sole inventor of the invention described and claimed in the specification X attached hereto or my/our patent Application Serial Number _____ filed _____ that as to common subject matter of this application and any earlier filed United States application of any above named to which I claim priority below, I do not believe that the same was ever known or used in the United States before my/our invention thereof or more than one year prior to such earlier application, or on public use or on sale in the United States more than one year prior to such earlier application(s), that the said common subject matter has not been patented before the date of such earlier application(s) in any country foreign to the United States on an application filed by me/us or my/our legal representatives or assigns more than twelve months prior to such earlier application(s) and that no application for patent on said common subject matter has been filed by me/us or my/our legal representatives or assigns in any country foreign to the United States prior to this application except those identified in item(s) below. If any, that as to any subject matter of this application which is not common to any said earlier application(s) (hereinafter subject matter), I am the first inventor, and do not believe that the same was ever known or used in the United States before my/our invention thereof or patented or described in any printed publication in any country before my/our invention thereof or more than one year prior to the date of this application, or on public use or on sale in the United States more than one year prior to the date of this application, and that said non-common subject matter has not been patented in any country foreign to the United States on an application filed by me/us or my/our legal representatives or assigns more than twelve months prior to the date of this application, and that no application for patent on said non-common subject matter has been filed by me/us or my/our legal representatives or assigns in any country foreign to the United States prior to this application except those identified in said item(s) below.

FULL NAME OF APPLICANT (INCL. AT LEAST ONE UNABBREVIATED FIRST OR MIDDLE NAME)

	LAST	FIRST	MIDDLE
2	Bradley	John	J.

RESIDENCE

	CITY	STATE (OR FOREIGN COUNTRY)	COUNTRY OF CITIZENSHIP
0	Garches	France	U.S.A.

POST OFFICE ADDRESS

	STREET ADDRESS	CITY	STATE (OR FOREIGN COUNTRY)	ZIPCODE
1	63 rue du 19 Janvier 92380	Garches	France	

FULL NAME OF APPLICANT (INCL. AT LEAST ONE UNABBREVIATED FIRST OR MIDDLE NAME)

	LAST	FIRST	MIDDLE
2	Franklin	Benjamin	S.

RESIDENCE

	CITY	STATE (OR FOREIGN COUNTRY)	COUNTRY OF CITIZENSHIP
0	Cambridge	Massachusetts	U.S.A.

POST OFFICE ADDRESS

	STREET ADDRESS	CITY	STATE (OR FOREIGN COUNTRY)	ZIPCODE
2	11 Peabody Terrace Apt. #2001	Cambridge	Massachusetts	02138

FULL NAME OF APPLICANT (INCL. AT LEAST ONE UNABBREVIATED FIRST OR MIDDLE NAME)

	LAST	FIRST	MIDDLE
2			

RESIDENCE

	CITY	STATE (OR FOREIGN COUNTRY)	COUNTRY OF CITIZENSHIP
0			

POST OFFICE ADDRESS

	STREET ADDRESS	CITY	STATE (OR FOREIGN COUNTRY)	ZIPCODE
3				

LISTING OF APPLICANTS CONTINUED ON PAGE 2 (if any) — YES X NO

TITLE OF INVENTION

3 8 1 — Switch System Base Mechanism

Earliest foreign application, if any, filed within 12 months prior to the U.S. filing date of this application of which priority is claimed under 35 USC 119

☐ YES ☐ NO

COUNTRY	APP. NO.	DATE OF FOREIGN FILING	CONTINUED ON PAGE 2 HEREOF
			YES X NO

☐ Specific reference to any prior related earlier filed U.S. application(s) of which this application is a CONTINUATION / CONTINUATION-IN-PART / DIVISION is made on page ____ of the specification for the purpose of receiving benefit of said earlier filing date(s) 35 USC 120

I/We hereby appoint the following as my/our X attorney(s) agent(s) to prosecute this application and transact all business in the Patent Office connected therewith

NAME	REGISTRATION NUMBER	ASSOCIATE POWER OF ATTORNEY OR AUTHORIZATION OF AGENT ATTACHED
Nicholas Prasinos Ronald T. Reiling	24,791 20,607	YES X NO

SEND CORRESPONDENCE TO

NAME	STREET	CITY AND STATE	ZIP CODE	TEL. NO.
Ronald T. Reiling Honeywell Information Systems Inc.	200 Smith Street	Waltham, Massachusetts	02154	(617) 890-8400, Ext. 42126

I/We further declare that all statements made herein of my/our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code, and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon

SIGNATURE(S) OF APPLICANT(S)	X HEREIN *	OR PAGE 2 ATTACHED *	DATE OF SIGNATURE
John J. Bradley Signature of Inventor			8 April 75 Date

* (IF PAGE 2 ATTACHED, SIGNED AND MADE A PART HEREOF) YES X NO

* (IF NOT SIGNED ON THIS PAGE, THIS DECLARATION IS NOT VALID UNLESS PAGE 2 IS SIGNED)

DRAWING
WAS ORIGINALLY FILED

570331

FIG. 1

S?0331

PRINT OF DRAWING
AS ORIGINALLY FILED

FIG. 2

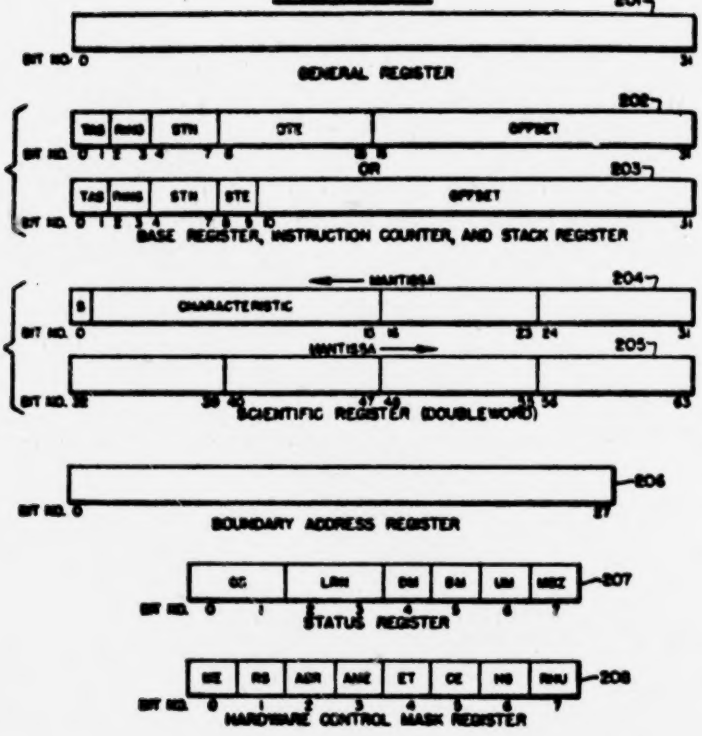


FIG. 3

LEGEND FOR RESERVED AREAS OF STORAGE IN REGISTERS

AME • ACCOUNTING MECHANISM
ABR • AUTO STORAGE RECONFIGURATION
BM • BINARY MASK
CC • CONDITION CODE
CE • CORRECTED ERROR
DM • DECIMAL MASK
ET • ENVIRONMENT TOLERANCE
HS • HARDWARE GATE
LRN • LAST RING NUMBER
MBZ • MUST BE ZERO
ME • MACHINE ERROR
RING • RING NUMBER
RHU • RESERVED FOR HARDWARE USE
RS • RETRY SUCCESSFUL
S • SIGN
STE • SEGMENT TABLE ENTRY
STN • SEGMENT TABLE NUMBER
TAG • DESCRIPTOR TYPE
UM • UNDERFLOW MASK

(AS ORIGINALLY FILED)

570331

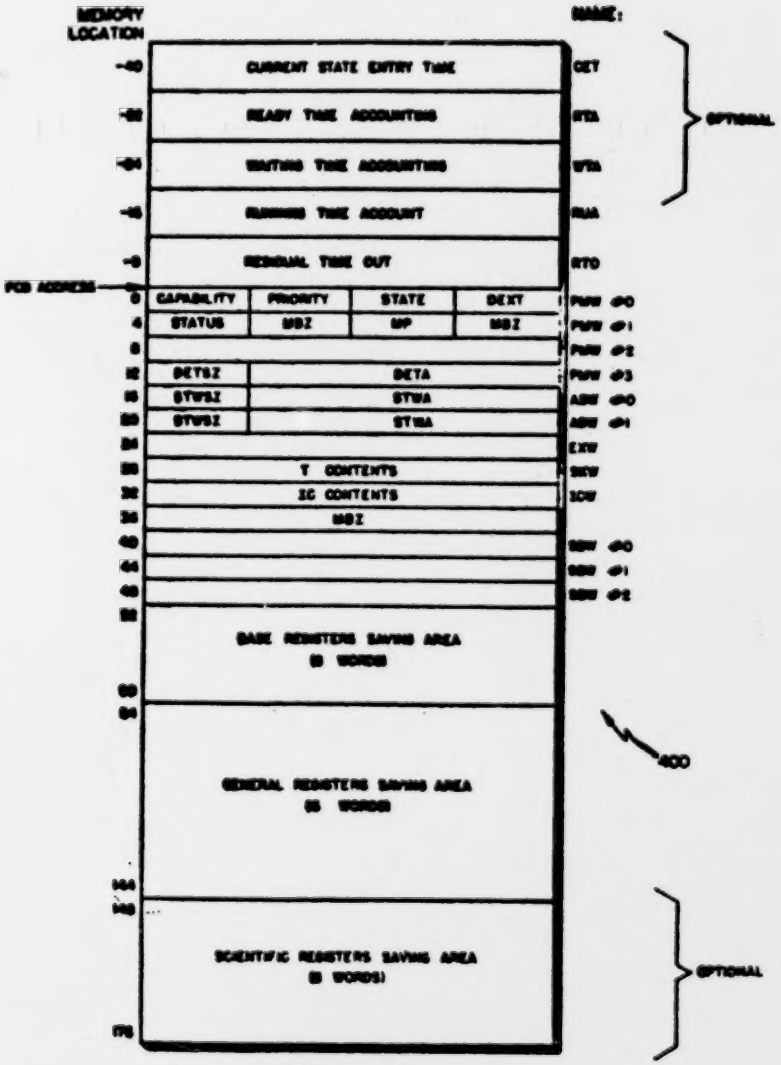


FIG. 4

570371

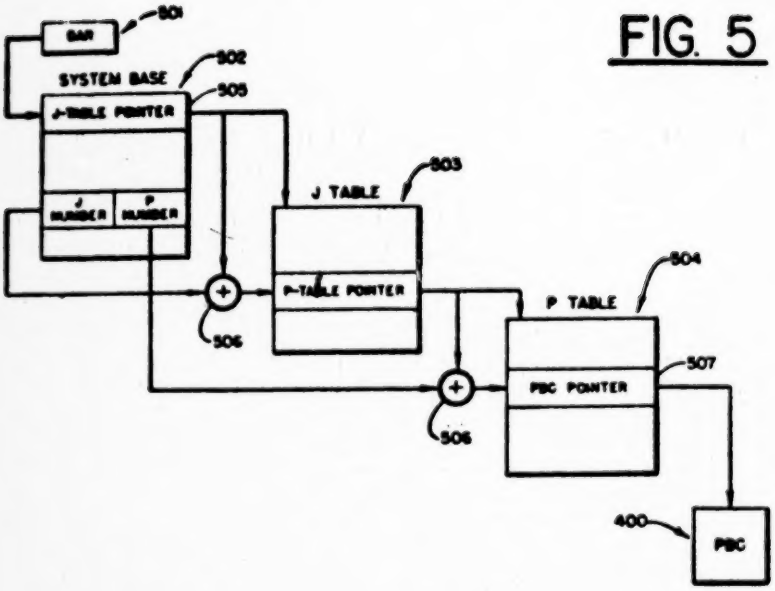


FIG. 5

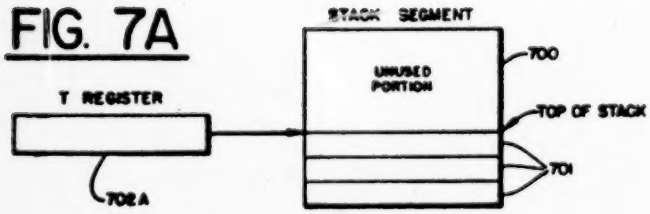


FIG. 7A

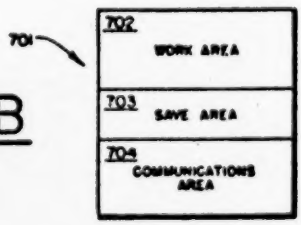
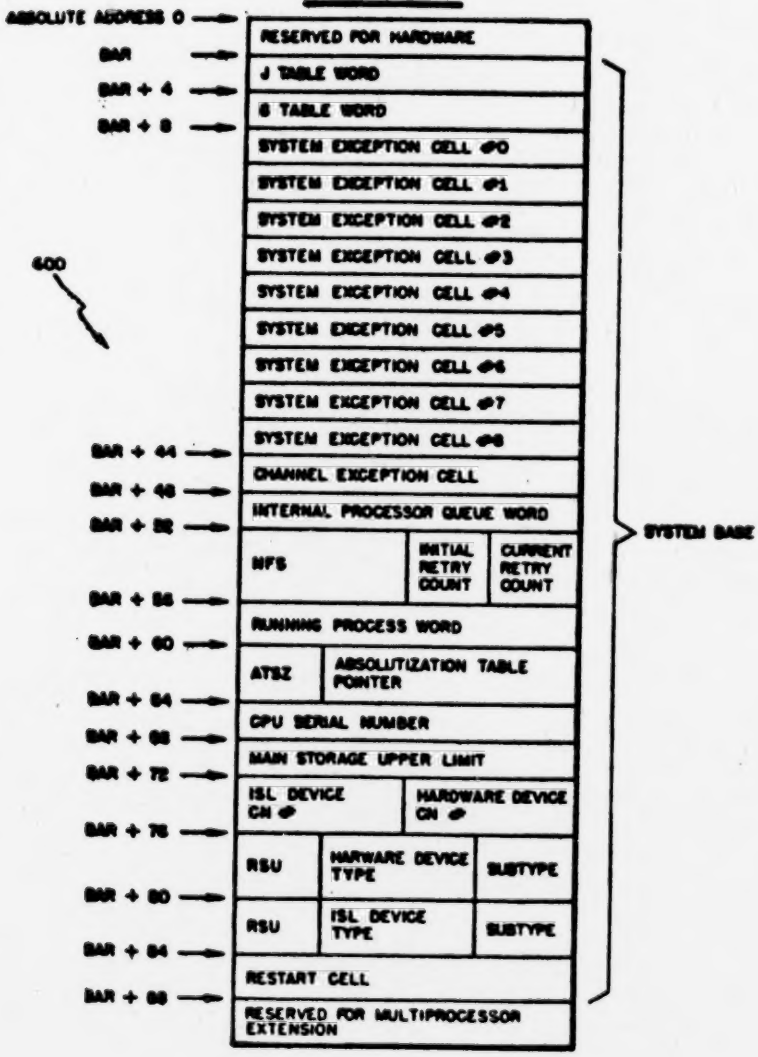


FIG. 7B

(48 ORIGINALLY FILED)

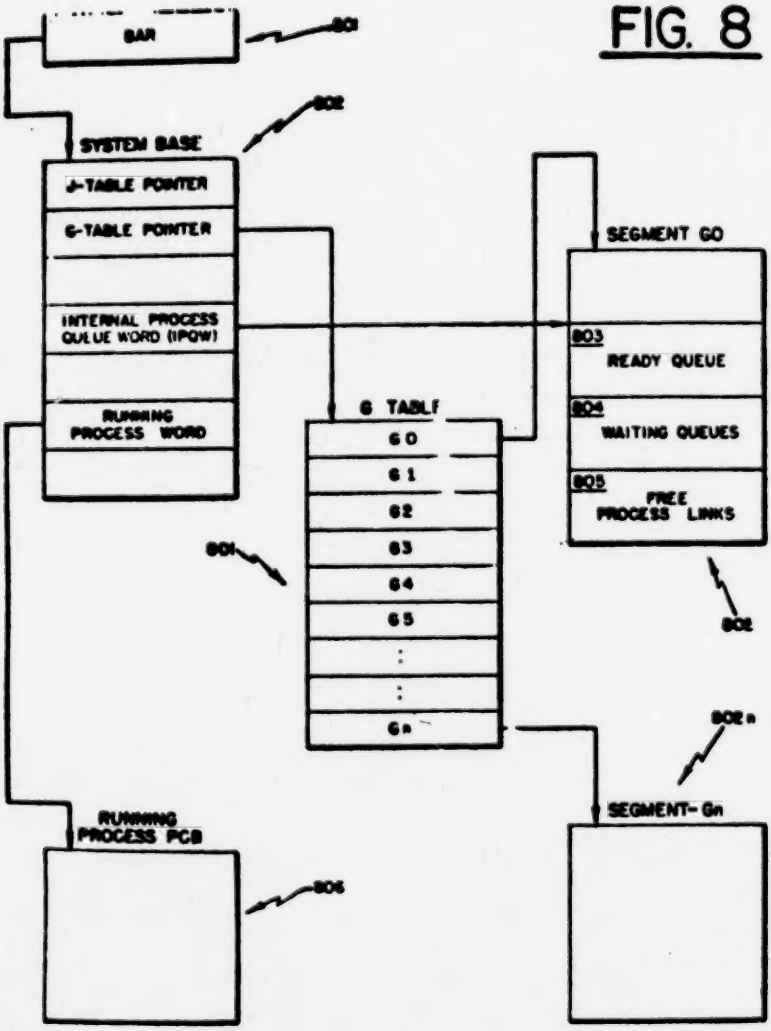
FIG. 6



AS ORIGINALLY FILED

57C331

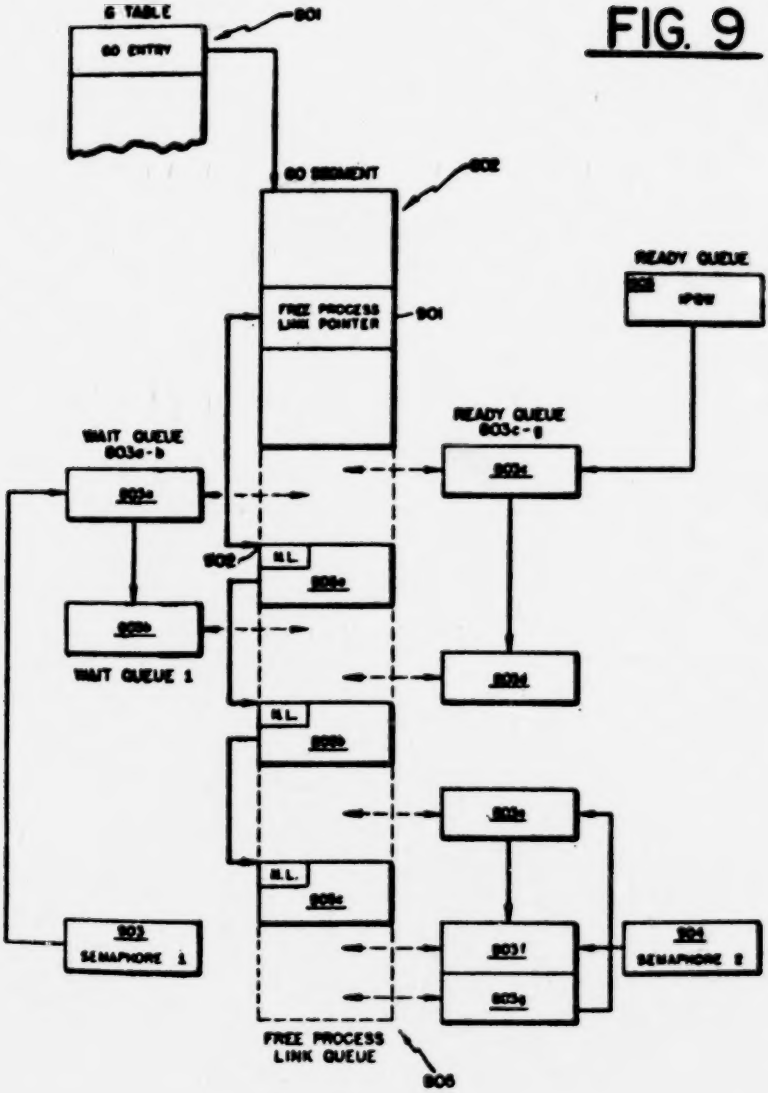
FIG. 8



PRINT OF DRAWING AS ORIGINALLY FILED

570378

FIG. 9



AS ORIGINALLY FILED

57C371

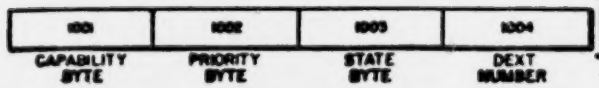


FIG. 10a

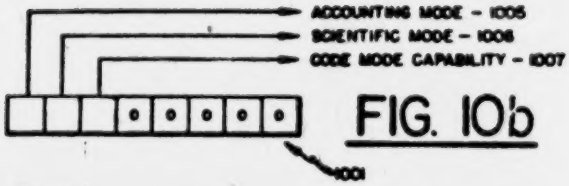


FIG. 10b

FIG. 10c

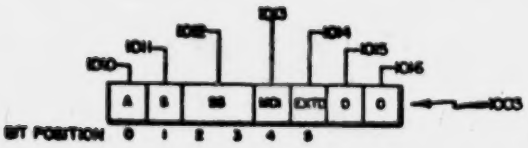


FIG. 10d

FIG. 10e

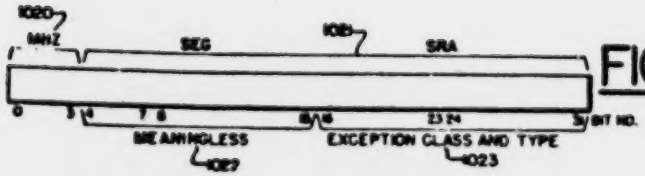


FIG. 10f

AS ORIGINALLY

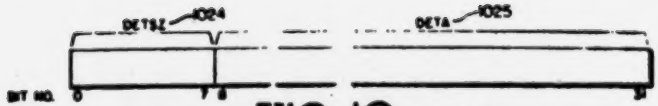


FIG. 10g

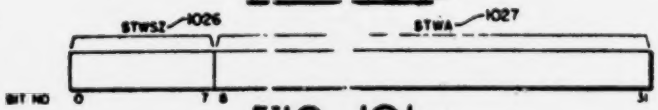


FIG. 10h

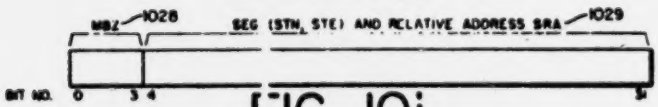


FIG. 10i

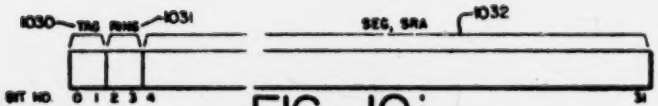


FIG. 10j

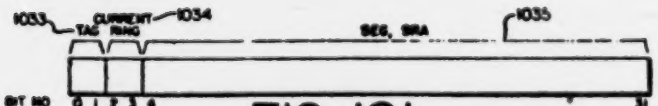


FIG. 10k

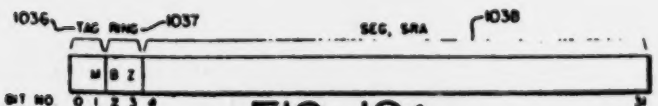


FIG. 10l

AS ORIGINALLY FILED

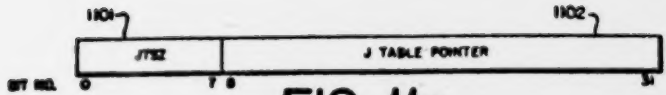


FIG. 11a

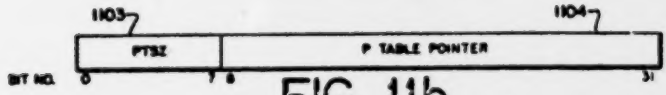


FIG. 11b

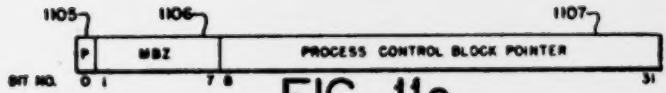


FIG. 11c

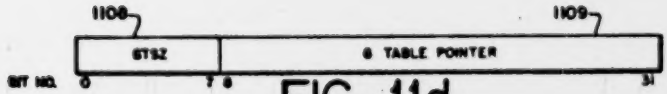


FIG. 11d

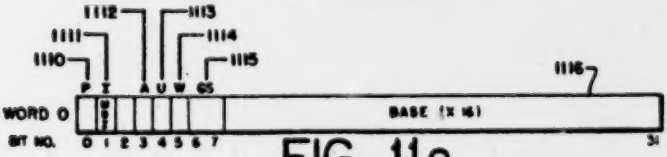


FIG. 11e

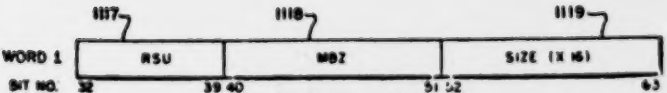


FIG. 11f

COPY OF DRAWING
AS ORIGINALLY FILED

570371

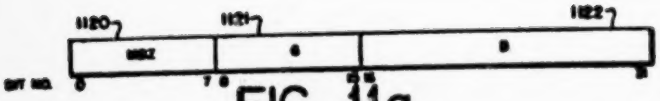


FIG. 11g

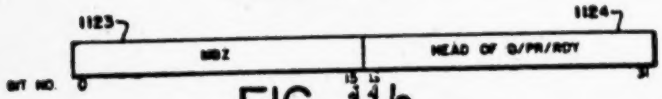


FIG. 11h

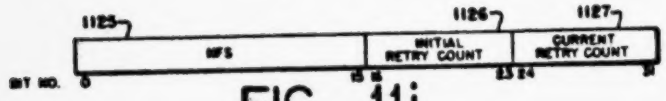


FIG. 11i

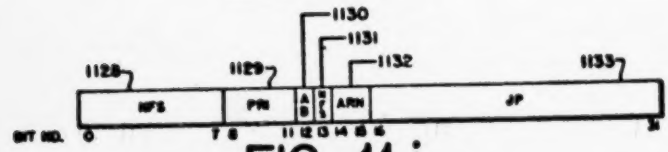


FIG. 11j

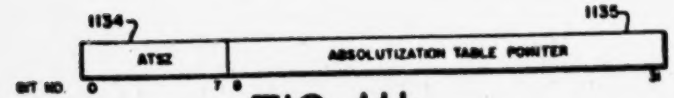


FIG. 11k

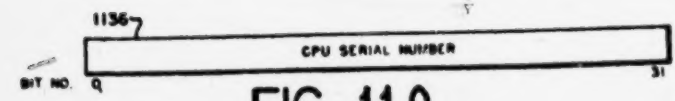


FIG. 11ℓ

PRINT OF DRAWING
NOT ORIGINALLY FILED

57:331

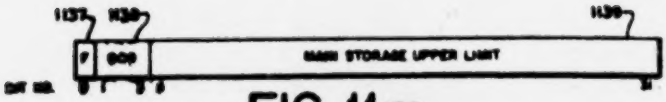


FIG. 11m

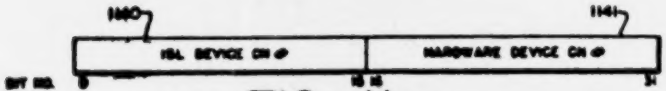


FIG. 11n

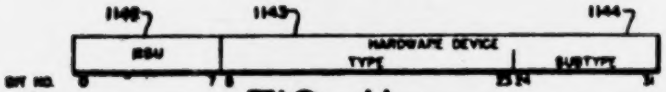


FIG. 11o

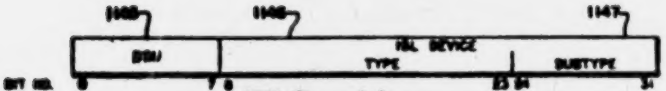


FIG. 11p

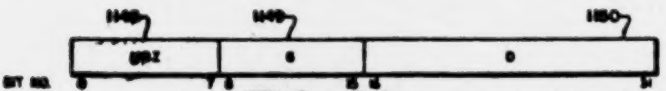


FIG. 11q

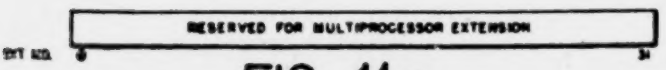
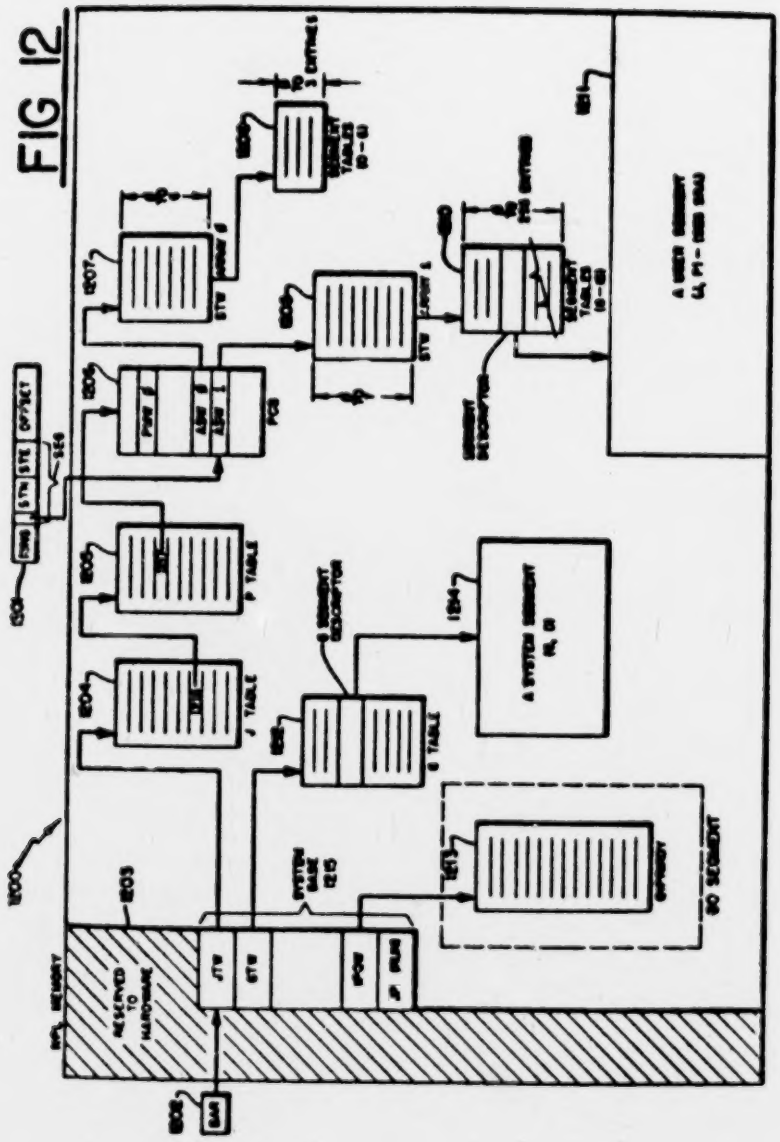


FIG. 11r

PATENT DRAWING
AS ORIGINALLY FILED

57C3'11

FIG 12



POINT OF DRAWING AS ORIGINALLY FILED

570371

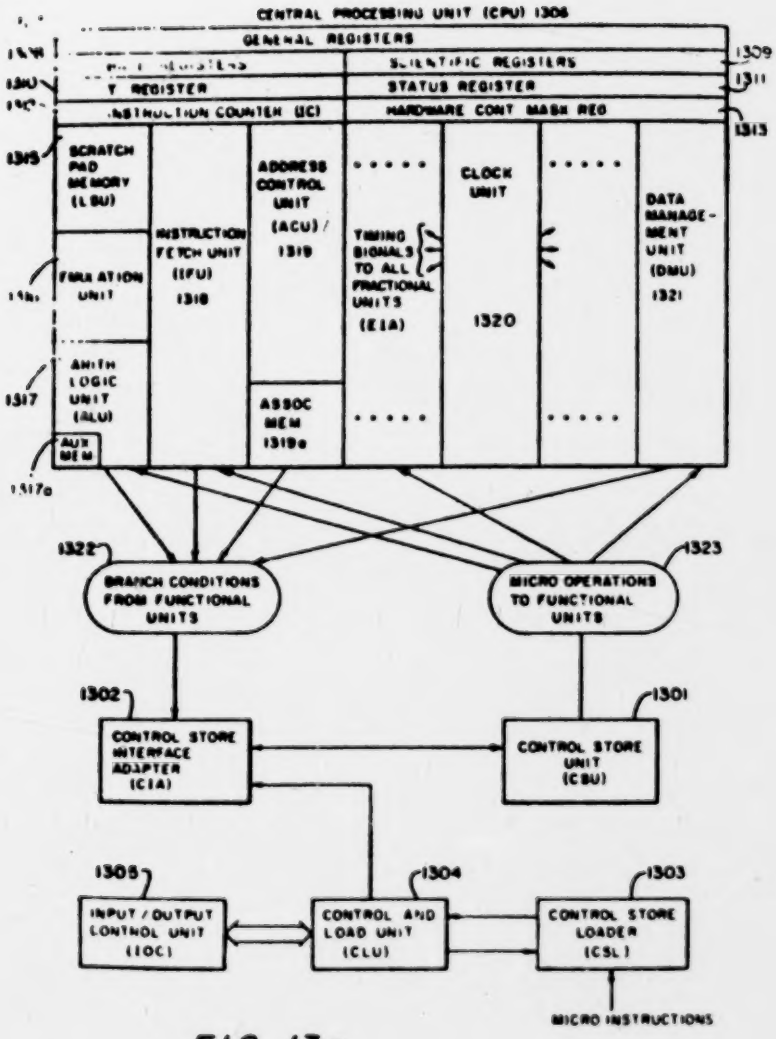


FIG. 13a

FRONT OF DRAWING
AS ORIGINALLY FILED

07C331

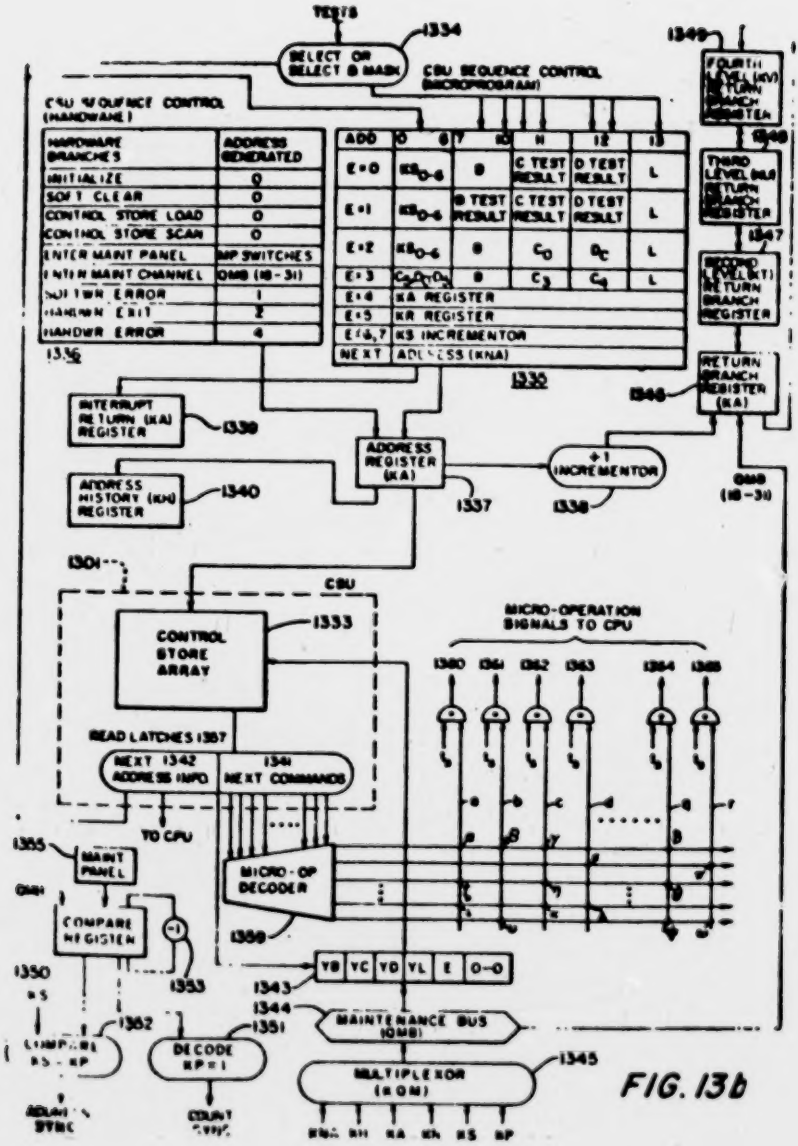


FIG. 13b

570331

PRINT OF DRAWING
WAS ORIGINALLY FILED

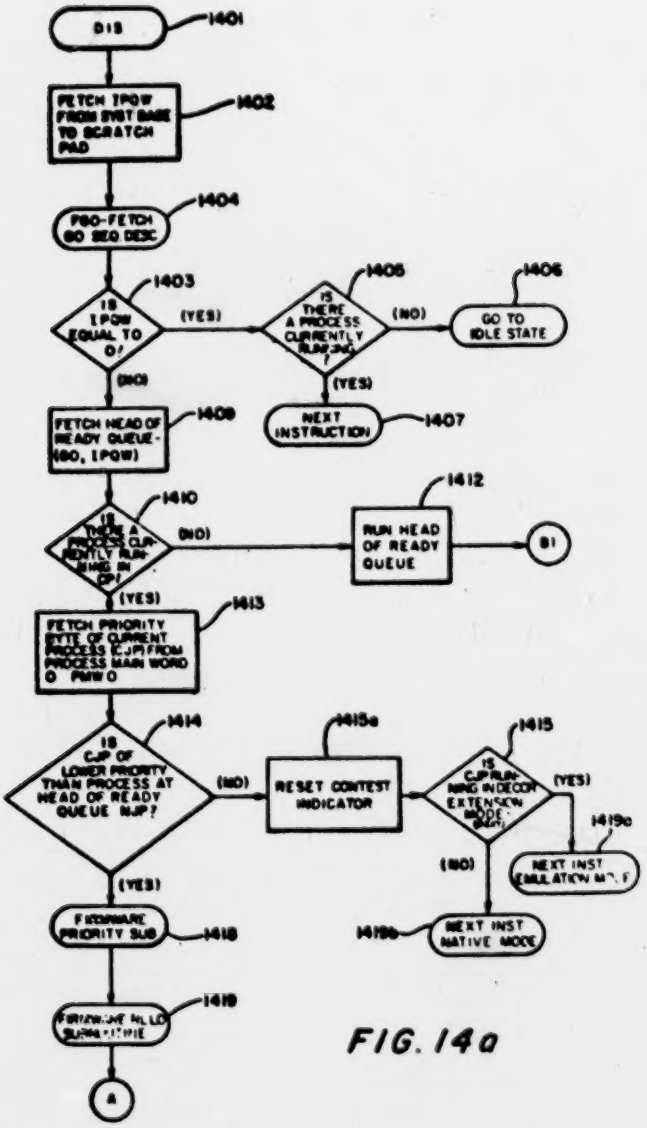


FIG. 14a

PRINT OF DRAWING
WAS ORIGINALLY FILED

57:331

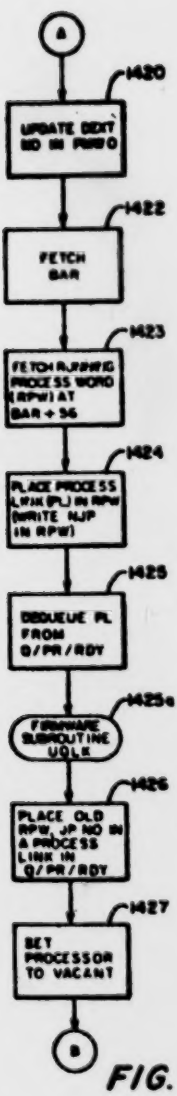


FIG. 14b

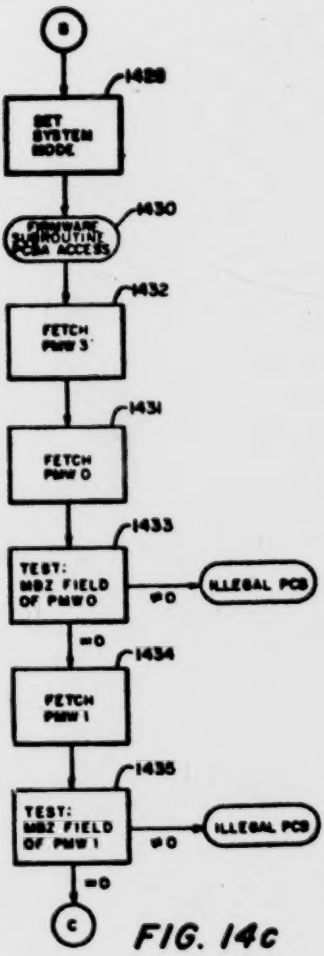


FIG. 14c

PRINT OF DRAWING
AS ORIGINALLY FILED

57C371

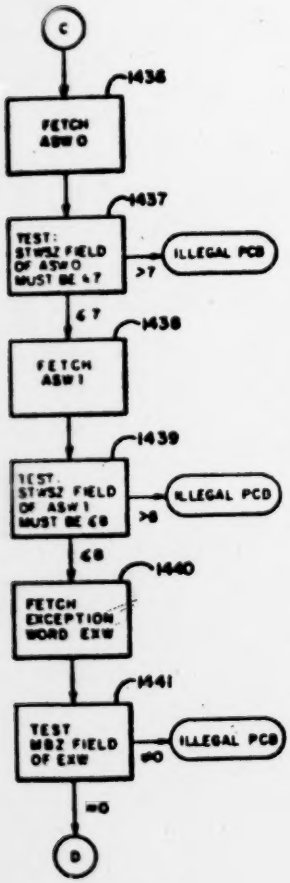


FIG. 14d

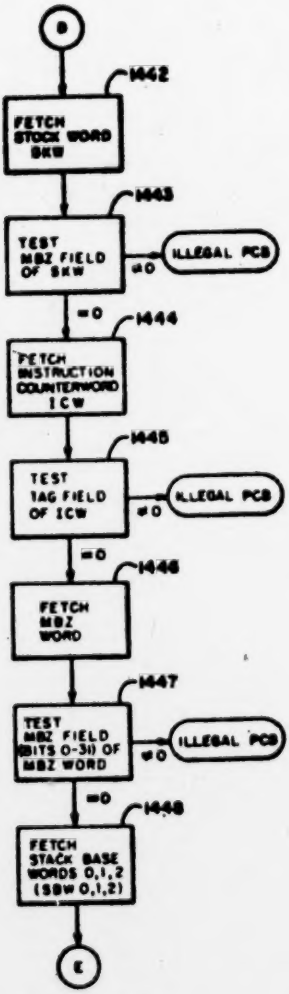


FIG. 14e

PATENT OF DRAWING
WAS ORIGINALLY FILED
570,131

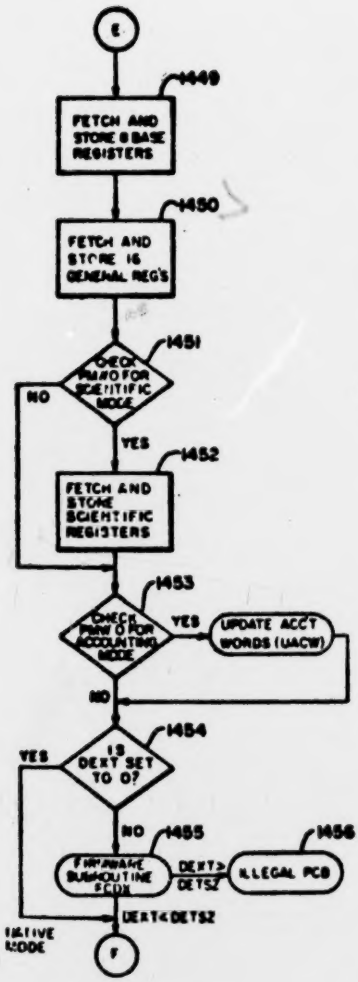


FIG. 14f

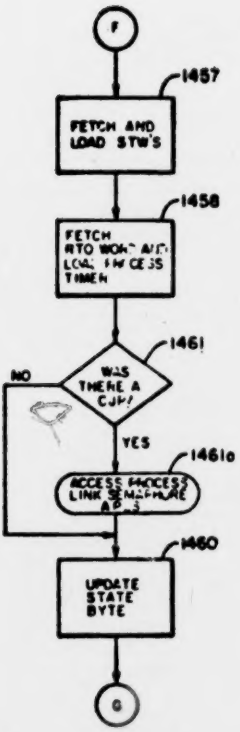


FIG. 14g

WAS ORIGINALLY FILED

570371

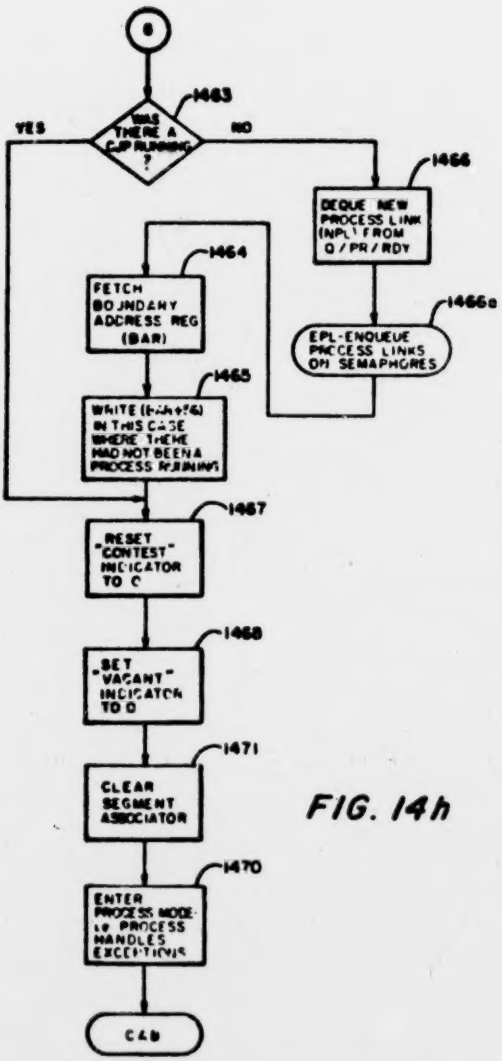


FIG. 14h

PATENT OF INVENTION
AS ORIGINALLY FILED

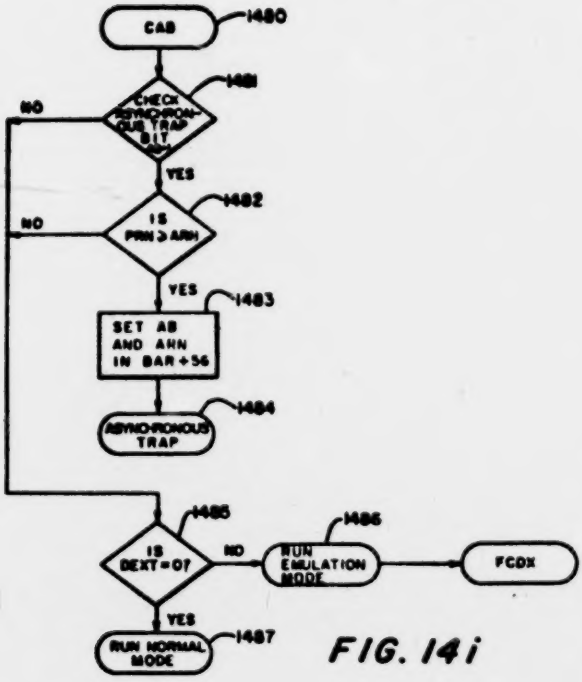


FIG. 14i

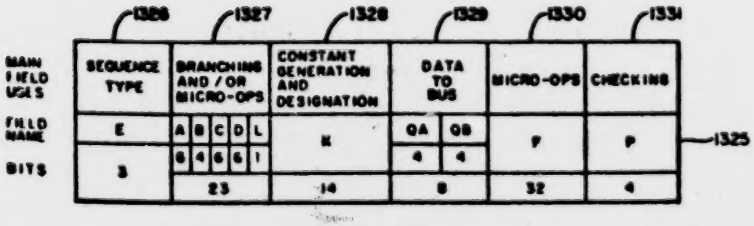


FIG. 13c

PRINT OF DRAWING AS ORIGINALLY FILED

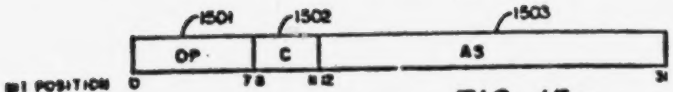


FIG. 15a

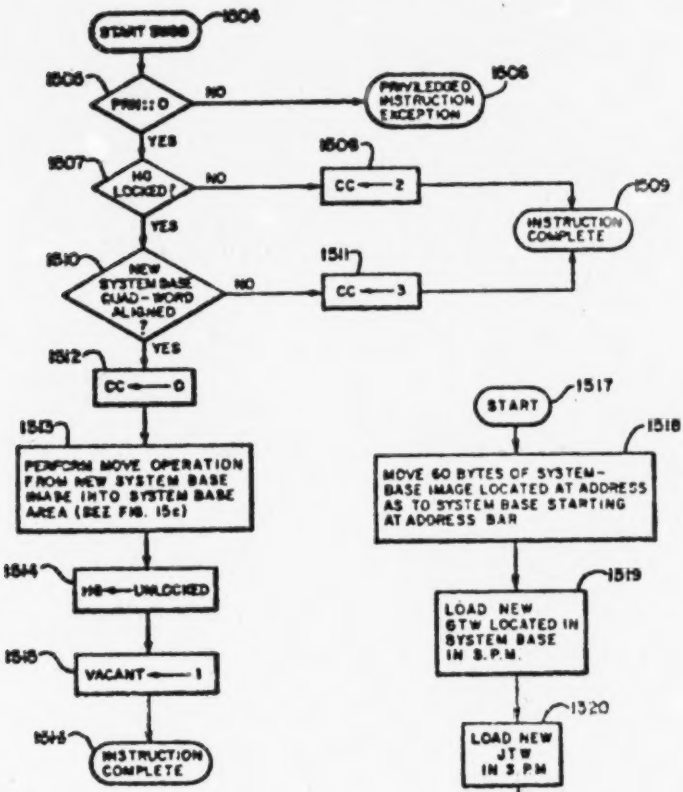


FIG. 15b

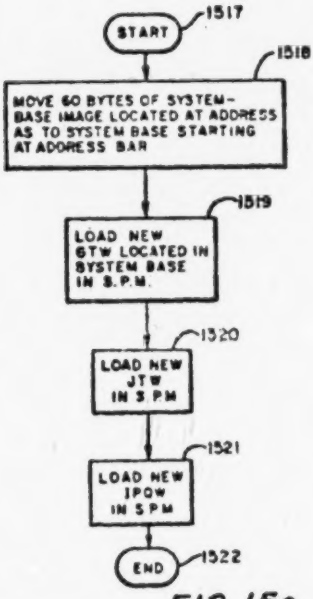


FIG. 15c

LETTER OF EXAMINER, MARCH 12, 1976.

☒ This application has been examined

Responsive to communication filed [illegible] ☐ this action is made final.

A SHORTENED STATUTORY PERIOD FOR RESPONSE TO THIS ACTION IS SET TO EXPIRE

THREE (3) MONTH(S) ~~XXXX~~ FROM THE DATE OF THIS LETTER.

PART I

The following attachments(s) are part of this action:

a. ☒ Notice of References Cited, Form PTO 892.

b. ☐ Notice of Informal Patent Drawing, PTO 948

c. ☐ Notice of Informal Patent Application, Form PTO 152.

d. ☐

PART II

Summary of Action

1. ☒ Claims 1 - 6 are presented for examination.

2. ☐ Claims _____ are allowed.

3. ☐ Claims _____ would be allowable if amended as indicated.

4. ☒ Claims 1 - 6 are rejected.

5. ☐ Claims _____ are objected to.

6. ☐ Claims _____ are subject to restriction or election requirement.

7. ☐ Claims _____ are withdrawn from consideration.

8. ☐ Since this application appears to be in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under Ex parte Quayle, 1935 C.D. 11, 453 OG. 213.

9. ☐ Since it appears that a discussion with applicant's representative may result in agreements whereby the application may be placed in condition for allowance, the examiner will telephone the representative within about 2 weeks from the date of this letter.

10. ☐ Receipt is acknowledged of papers under 35 USC 119, which papers have been placed of record in the file.

11. ☐ Applicant's claim for priority based on an application filed in _____ on _____ is acknowledged. It is noted, however, that a certified copy as required by 35 USC 119 has not been received.

12. ☐ Other

1) Due to the length of the specification it has not been checked to the extent of determining all minor errors. Applicants are requested to review the entire specification and correct any errors noted.

2) The Abstract of the Disclosure should be reviewed and corrected to more specifically and clearly state the character of the subject matter deemed to be new in the art.

3) In the specification, pages 2-4 and various others throughout the disclosure, application serial numbers and filing dates have been omitted.

In lines 1 and 2, page 2, the purpose of the procedure to allow incorporation by reference to previous patents and applications was to reduce the length of disclosure in related applications and simplify the examining and printing procedures involved. Applicants have incorporated by reference numerous applications and/or patents which do not appear to be "essential subject matter" relative to the disclosure of the instant invention. If these prior applications are to be referenced as related applications then it should be so indicated rather than incorporating all of them by reference.

4) The specification is objected to under 35 USC 112 as failing to set forth the subject matter of the invention in such terms as to enable one to make and use the invention without undue experimentation. The basic elements of the system are disclosed in no more detail than hollow rectangles and the complex interrelationship between the various elements is disclosed only in functional statements of desired results. For example, the control unit 130 of Figure 1 is stated to contain firmware which initializes the system, controls the CPU 104 and IOC 105, and encodes an instruction set. However, the elements of the control unit and the instructions for configuring the various elements have not been disclosed in sufficient detail to enable one of ordinary skill in the art to reconstruct the invention.

Applicants have referred to basic data structures on page 11 and following disclosure as being data formats,

software visible registers, and instruction formats. This use of the term "structures" to refer to data formats, etc. is carried over into the claims and results in confusion as to the nature of the invention. The subject matter disclosed in Figure 2 must be deemed in the nature of data format and not of the nature of apparatus as usually intended by use of the word "structure." Clarification is requested.

5(Claims 1-6 are rejected as drawn to subject matter which is inadequately disclosed, 35 USC 112, first paragraph.

6) Claims 1-6 are further rejected under 35 USC 112, second paragraph, as failing to particularly point out and distinctly claim the subject matter sought to be patented.

The structural connections and cooperative relationship between the various elements comprising the claimed combination have not been clearly established. Although the claims are recited as a combination of devices in the means plus statement of function language, most of the claim language does not recite "structure" in the sense of apparatus but rather uses "means" to refer to software or firmware instructions. Applicant is therefore requested to point out the structure in the drawings upon which the various "means" are to be read.

7) Claims 1-6 are further rejected under 35 USC 101 as drawn to non-statutory subject matter.

The Supreme Court in *Gottschalk* v. *Benson*, 175 USPQ 673, stated that Court's views on the patentability of certain claims which set forth an invention related to the processing of data by a programmed computer.

The Court in Benson addressed itself to the broad question of patentability of computer programs and stated that if such subject matter is to be patented, considerable programs are raised which only committees of Congress can manage. Justice Douglas, spe king for the Court, declined to decide such policy matters and refused to extend the patent statutes to embrace computer programming, absent considered action by the Congress.

8) Claims 1-6 are further rejected under 35 USC 102 as unpatentable over the Hauck patent. The apparatus in Figure 1 of Hauck cooperates to perform the function of applicant's claimed invention.

9) The additional art cited, although not applied against the claims, is considered pertinent in showing related systems.

10) All claims have been rejected. This application should be put in condition for final disposition.

/s/ Charles E. Atkinson
CHARLES E. ATKINSON
EXAMINER
GROUP ART UNIT 236

NOTICE OF REFERENCES CITED.

U.S. PATENTS

*		PATENT NO.	DATE	PATENTEE	CLASS	SUB-CLASS	FILING DATE IF APPROPRIATE
	A	3,651,473	3/1972	Faber	340	172.5	
	B	3,665,487	5/1972	Campbell et al.	340	172.5	
	C	3,670,307	6/1972	Arnold et al.	340	172.5	
	D	3,735,364	5/1973	Hatta et al.	340	172.5	
	E	3,725,876	4/1973	Hauck	340	172.5	
	F	3,786,427	1/1974	Schmidt et al.	340	172.5	
	G	3,786,813	1/1974	Cole et al.	340	172.5	
	H	3,889,243	6/1975	Drimak			
	I						
	J						
	K						

FOREIGN PATENTS OR PUBLISHED APPLICATIONS

	PATENT NO.	DATE	COUNTRY	NAME	CLASS	SUB-CLASS	PERTINENT SHTS. DWG	PERTINENT PP. SPEC.
L								
M								
N								
O								
P								
Q								

* OTHER REFERENCES (Including Author, Title, Date, Pertinent Pages, Etc.)

R

S

T

U

EXAMINER	DATE	
C.E. Atkinson	12/4/75	☐ CHECK HERE IF FOREIGN EXCHANGE

* ATTENTION R.O.B.: Do not furnish copy of this box is checked.

AMENDMENT, AUGUST 11, 1976.

August 10, 1976

Commissioner of Patents & Trademarks
Washington
District of Columbia 20231

Dear Sir:

Responsive to the Office Action of 3/12/76 the due date of which was extended two months from 6/12/76 to and including 8/12/76, please amend the above-identified application as follows:

IN THE SPECIFICATION

Page 1, line 2, delete line "A model-idenpendent instruction in firmware" and substitute therefore—Firmware-

Page 1, line 6, start a new paragraph as follows:

Hardware structures centralled by a microprogrammed central store, which provides for the generation of signals which cause the creation of an image in main memory of a communication center comprised of registers at another location. The microprogrammed control store further generates signals which causes the interchange of the information between the communication center and its image.

Page 2, line 1, delete "incorporated by reference" and substitute therefor—related—.

Page 2, line 23, after "invented by" insert—Pravinsinh L. Parmar, et al—.

Page 2, line 24, delete "the same date as the instant invention" and substitute therefor—December 12, 1973—.

Page 2, line 25, before "and assigned" insert 424,239 and issued into Patent No. 3,916,385 on 10/28/75,.

Page 2, line 26, after "invented by" insert—Jacques Bienvenu—.

Page 2, line 27, after "filed on" insert—5/16/74—.

Page 2, line 26, after "Number" insert—470,496—.

Page 2, line 29, delete "Information in a Multiprogram Multiprocessor Computer System" and substitute therefor Data in an Information Multiprocessing System by Implementing a Concept of Rings to Represent the Different Levels of Privileges Among Processes.

Page 2, line 30, after "invented by" insert—Marc Appell—.

Page 2, line 30, after "filed on" insert—December 2, 1974—.

Page 2, line 31, after "Serial No." insert—528,953—.

Page 3, delete lines 1-3 as follows—"Ring Crossing Mechanism" invented by ————, filed on ———— and having Serial No. ———— and assigned to the same assignee named herein.—.

Page 3, line 4, delete "Operations" and substitute therefor—Mechanism—.

Page 3, line 5, before "filed on" insert—Marc Appell—.

Page 3, line 8, after "filed on" insert—December 3, 1974—.

Page 3, line 8, after "Number" insert—529,256—.

Page 3, line 10, after "Resources" insert—To Job Step in a Multiprogram Computer—.

Page 3, line 10, after "invented by" insert—Jean-Pierre Onimus—.

Page 3, line 11, after "filed on" insert—December 2, 1974—.

Page 3, line 11, after "Number" insert—529,255—.

Page 3, delete lines 13-16 as follows—"Data Processing Memory Management System" invented by ———— ————, filed on ———— and having Serial Number ———— and assigned to the same assignee named herein.—.

Page 3, line 17, delete "Computer" and substitute therefor—Data Processing.

Page 3, lines 17-18, delete "Controlled by Reloadable Site-Dependent Information Structures".

Page 3, line 19, before "filed on" insert—William Woods—.

Page 3, line 19, after "filed on" insert—December 2, 1974.

Page 3, line 20, before "and assigned" insert—528,-814—.

Page 3, line 21, after "invented by" insert—Claude Carre—.

Page 3, line 23, before ", and having" insert—December 2, 1974—.

Page 3, line 22, after "Number" insert—528,951—.

Page 3, delete lines 24-27 as follows—"Absolutization During Initialization", invented by ————, filed on ————, and having Serial Number ———— and assigned to the same assignee named herein—.

Page 4, line 4, after "invented by" insert—Claude Carre—.

Page 4, line 5, after "filed on" insert—December 2, 1974—.

Page 4, line 5, after "Number" insert—528,951—.

Page 4, line 11, delete "Absolutization During Initialization" substitute therefor—Arrangement for Initializing a Data Processing System—.

Page 4, line 12, after "invented by" insert—William Woods—.

Page 4, line 12, after "filed on" insert—December 2, 1974—.

Page 4, line 13, after "Number" insert—528,814—.

Page 6, line 2, after "application" insert—references 12 and 14 on page 3—.

Page 12, line 21, after "No." insert—470,496—.

Page 12, line 22, before "entitled" insert—5/16/74—.

Page 17, line 31, after "No." insert—528,953—.

Page 17, line 31, after "Filed" insert—December 2, 1974—.

Page 18, line 1, delete "Information in a Multiprogram, Multiprocessor Computer System" and substitute therefor Data in an Information Multiprocessing System by Implementing the Concept of Rings to Represent the Different Levels of Privilege.

Page 18, lines 2-4, delete—; also see patent application no. ————, filed ————, entitled "Ring Crossing Mechanism" both ————.

Page 19, line 14, after "no." insert—470,496—.

Page 19, line 14, after "filed" insert—5/16/74—.

Page 19, line 15, delete "; also see patent no." and substitute therefor—.—.

Page 42, line 14, after "No." insert—470,496—.

Page 42, line 14, after "filed on" insert—5/16/74—.

Page 54, line 24, after "No." insert—528,814—.

Page 54, line 24, delete "System Initialization Procedure" and substitute therefor—Arrangement for Initializing a Data Processing System—.

Page 54, line 26, delete "Start and Suspend Instruction" and substitute therefor—Process Management Instruction for a Central Processor—.

Page 54, line 29, after "No." insert—529,017 filed on December 2, 1974—.

Page 54, lines 29 and 30 delete "Start and Suspend Instruction" and substitute therefor "Process Management Instruction for a Central Processor" also assigned to the same assignee named herein.

Page 55, line 4, after "No." insert—529,256 filed on 12/2/74—.

Page 55, lines 4 and 5, delete "P and V Operations of Semaphores" and substitute therefor—Process Synchronization Utilizing Semaphores—.

Page 64, line 6, delete "16" and substitute therefor—15—.

Page 79, between lines 5 and 6 insert Appendix I as follows:

APPENDIX I

The following information is provided as background information to better understand the invention, although such information is not necessary to practice the invention.

Further Information with Respect to Figure 1:

(1) Emulators (112)

(a) Emulators are found in the IBM system/370. See page 265 of a book entitled *Computer Organization and the System/370*, by Harry Katzan, Jr., in 1971 by Van-Nostrand Reinhold Company of New York, New York.

(b) Emulators are also described in the following U.S. patents:

3,374,466 issued 3/19/68;
3,543,245 issued 11/24/70;
3,698,007 issued 10/10/72;
3,721,961 issued 3/20/73.

(2) Computation Unit (111)

(a) Computational units are well known in the art and are also known as Arithmetic and Logic Unit (ALU). See the following U.S. patents:

3,400,371 issued 9/3/68;
3,404,378 issued 11/1/68;
3,238,508 issued 3/1/66;
3,201,762 issued 8/17/65.

(3) Control Unit (130)

(a) Typical control units are described in a book entitled *Microprogramming: Principles and Practices*, by Samir S. Husson, published in 1970 by Prentice-Hall Inc., of Englewood Cliffs, New Jersey; see also a book entitled *Computer Organization and Microprogramming*, by Yao-Yaohan Chu published in 1972 by Prentice-Hall Inc., of Englewood Cliffs, New Jersey. Typical state of the art control units are disclosed in the following U.S. patents:

3,400,371 issued 9/3/68;
3,370,274 issued 2/20/68;
3,380,025 issued 4/23/68;
3,268,872 issued 8/23/66;
3,302,183 issued 1/31/67;
3,391,394 issued 7/2/68;
3,302,183 issued 1/31/67;
3,477,063 issued 11/4/69;
3,391,394 issued 7/2/68;
3,646,522 issued 2/29/72.

(4) Buffer Store (110)

(a) See reference 1, page 2, of this application.

(b) See also pages 296-311 of the above referenced book on *Computer Organization and Microprogramming*, by Yao-Yaohan Chu.

(c) Typical buffer stores are also disclosed in the following U.S. patents:

3,411,142 issued 11/12/68;
3,609,665 issued 9/28/71;
3,508,220 issued 4/21/70.

(5) Main Memory Synchronizer (109)
(a) See reference 4, page 2, of this application
(b) See also U.S. patent:

3,449,723 issued 6/10/69.

(6) Associative Memory (132)
(a) Associative memories are well known in the prior art, see pages 273-284 of the above referenced book on *Computer Organization and Microprogramming*, by Yao-Yaohan Chu; see also the following U.S. patents:

3,648,254 issued 3/7/72;
3,456,243 issued 7/15/69;
3,253,265 issued 5/24/66;
3,248,704 issued 4/26/66;
3,241,123 issued 3/15/66;
3,170,142 issued 2/16/65.

(7) Address Control Unit (ACU) 131
(a) See reference 6, page 2, of this application.
(8) Input/Output Control Unit (105)
(a) Typical I/O control units are disclosed in the following U.S. patents:

3,559,187 issued 1/26/61;
3,593,299 issued 7/13/71;
3,251,040 issued 5/10/66;
3,573,741 issued 4/6/71;
3,444,520 issued 5/13/69;
3,680,054 issued 7/25/72;
3,359,543 issued 12/19/67;
3,651,480 issued 3/21/72;
3,248,701 issued 4/26/66;
3,254,330 issued 5/31/66.

(9) Peripheral Control Unit (106)
(a) See Chapter 10 of the above referenced book on *Computer Organization and Microprogramming*, by Yao-Yaohan Chu.

(b) Typical peripheral control units including device adapters are further disclosed in the following U.S. patents:

3,742,457 issued 6/26/73;
3,753,236 issued 7/14/73;
3,408,632 issued 10/29/68;
3,614,740 issued 10/19/71.

(10) Device Adapters (107)

(a) See reference 9, of this page. See also the following U.S. patents:

3,573,741 issued 4/6/71;
3,680,054 issued 7/25/72.

(11) Main Memory (102)

(a) See references 2-7, inclusive on page 2 of this application.

Background Information Relative to Figure 13a

In addition to the references already cited in this Appendix, the following are also included:

(12) General Registers (1307-1311)

(a) Registers are well known in the computer art. Typical ones are disclosed in the following U.S. patents:

3,117,307 issued 1/7/64;
3,108,193 issued 10/22/63;
3,383,521 issued 5/14/68;
3,289,010 issued 11/29/66;
3,258,696 issued 6/28/66.

(13) Hardware Control Mask Register (1313)

(a) See also U.S. patents:

3,351,915 issued 11/7/67;
3,602,898 issued 8/31/71.

(14) Instruction Counter (1312)

(a) Instruction counters are well known in the computer art. See for example U.S. patent:

3,162,841 issued 12/22/64.

(15) Scratch Pad Memory (1315)

(a) Typical Scratch Pad Memories are disclosed in the following U.S. patents:

3,248,709 issued 4/26/66;
3,351,909 issued 11/7/67.

(16) Instruction Fetch Unit (1319)

(a) Instruction fetch units are well known in the computer art. See for example pages 339-347 of the above referenced book on *Microprogramming: Principles and Practices*, by Samir S. Husson.

(b) Typical instruction fetch units are also disclosed in the following U.S. patents:

3,609,700 issued 9/28/71;
3,614,747 issued 10/19/71.

(17) Clock Unit and Timing Circuits (1320)

(a) The computer art is replete with clock and timing circuits. The following U.S. patents disclose typical clock and timing circuits:

3,355,710 issued 11/28/67;
3,417,379 issued 12/17/68;
3,254,329 issued 5/31/66;
2,297,305 issued 3/1/66;
2,781,446 issued 2/12/57;
3,413,454 issued 11/26/68;
3,555,594 issued 11/28/67.

(18) Data Management Unit (1321)

(a) See the following U.S. patents for typical data management and control systems:

3,618,045 issued 11/2/71;
3,665,487 issued 5/23/72;
3,639,912 issued 2/1/72;
3,495,220 issued 2/10/70;
3,599,158 issued 8/10/71.

(19) Control Store Unit (1301)

(a) Control stores are well known in the computer art. See prior reference books entitled *Microprogramming: Principles and Practices*, by Samir S. Husson and Com-

puter Organization and Microprogramming, by Yao-Yaohan Chu.

(b) The following U.S. patents also disclose typical control stores:

3,380,025 issued 4/23/68;
3,469,247 issued 9/23/69;
3,160,858 issued 12/8/64.

(20) Control Store Interface Adapter (1302)

(a) See the following U.S. patents:

3,636,524 issued 1/18/72;
3,353,162 issued 11/14/67;
3,868,649 issued 2/25/75;
3,611,311 issued 11/5/71.

(21) Control Store Loader and Control and Load Unit (1303, 1304)

(a) Program and microprogrammed loaders are well known in the computer art. See for example pages 284-294 of the above-referenced book on *Computer Organization and Microprogramming*, by Yao-Yaohan Chu.

(22) Firmware Computer

(a) The following U.S. patents further disclose microprogrammed computer processors including microprogrammed control units with which the instant invention may be practiced:

3,400,371 issued 9/3/68;
3,656,123 issued 4/11/72.

IN THE CLAIMS

* * * *

For Claims 1, 2 and 4, see Rejected Claims 1, 2 and 4.

REMARKS

Claims 1-6 are still active in the application.

Claims 1, 2 and 4 have been amended.

Reconsideration of the instant application is respectfully requested.

The Applicants have made minor amendments to the specification to correct obvious errors and to complete the information on related applications. Claims 1, 2 and 4 have also been amended for overcoming the Examiner's objection under 35 USC 112. Moreover, Appendix I has been supplied to the specification by amendment. Appendix I supplies the prior art background information which the Examiner has labeled as "hollow rectangles" . . . and "not disclosed in sufficient detail to enable one of ordinary skill in the art to reconstruct the invention." It should be noted that all of this information is background information and pertains mainly to Figures 1, 13a and 13b wherein prior art devices or their equivalents are identified or correspond to the blocks in those figures having the same reference numeral in the Appendix as in the drawing are provided. Since this matter is totally background information and is not necessary to the practice of the invention by a person of ordinary skill in the art without undue experimentation, it entails no new matter with respect to the invention. Accordingly, it is submitted that the Applicants are entitled to have it included with the original specification.

The following arguments are further submitted for the Examiner's consideration relative to the 35 USC 112 insufficiency disclosure. The Applicants have followed 35 USC 112 and established case law which interprets the requirements of 35 USC 112 in making their disclosure. Several cases have spoken with respect to the disclosure. However, before discussing any of these cases, it should be pointed out that the Board of Appeals of the Patent Office has also spoken with respect to the disclosure of complex system inventions In Ex parte Billottet decided 1/26/76 by the Board of Appeals, the Board recognized functional-type block diagrams as a means for disclosing a complex system invention even though the specification does not particularly identify each element represented by the blocks and does not specify particular apparatus intended to carry out each function. The use of block diagrams to disclose the invention is sufficient to comply with the best mode requirements of 35 USC 112, first paragraph, and the requirements of Section

112, second paragraph, to distinctly point out the invention if the individual components of the system are conventional and the invention resides in the particular arrangement of these components.

The Board said, "Block diagrams may be acceptable and, in fact, preferable in cases of this nature where they serve in conjunction with the rest of the specification to enable one skilled in the art to practice the claimed invention. . . drawings and descriptions of circuits that are conventional would in a complex system disclosure such as is here involved, tend to obscure the claimed invention. Here we feel the practice of the claimed invention could be achieved without the necessity of further experimentation itself of an inventive nature."

In In Re Ghiron and Ulrich, 169 USPQ 723, the Court said "Functional-type block diagrams are preferable disclosure if they serve in conjunction with the rest of the specification to enable persons skilled in the art to select prior art apparatus that will produce the results required to practice the claimed process and to practice claimed invention with only a reasonable degree of routine experimentation; with respect to the best mode, the Court said in In Re Bosie, 149 USPQ 789 "Inventor complies with best mode requirement of 35 USC 112 if he does not conceal what he feels is a preferred embodiment of the invention; there is no statutory basis for reading into best mode portion of the statute, a requirement that the mode disclosed be in fact the optimum mode of carrying out invention; *it is error to require that the best mode set forth details not relating to the essence of invention*; there is no basis for best mode rejection merely because the Applicant does not pin-point within the specification some portions that are termed best mode."

Additionally, the Examiner's attention is drawn to U.S. Patent No. 3,891,974 issued June 24, 1975 and assigned to Honeywell Information Systems Inc. In this patent, Figures 1-14h are identical. Moreover, the specification describing these figures is substantially the same. Accordingly, it is submitted that the Patent Office has already placed the stamp of approval on the disclosure pertaining to Figures 1-14H among which are the blocks

the Examiner is questioning. There is therefore, precedence in the Patent Office itself which has recognized as sufficiently disclosed at least those items that the Examiner is questioning in the instant Office Action. For example, the Examiner questioned "the control unit 130 of Figure 1 (which) is stated to contain firmware which initializes the system, controls the CPU 104 and IOC 105, and includes an instruction set. However, the elements of the control unit and the instructions for configuring the various elements have not been disclosed in sufficient detail to enable one of ordinary skill in the art to reconstruct the invention."

The Applicants submit that before it can be established that Applicants have failed to meet the best mode and enabling requirements of 35 USC 112, it must be first determined that which is the subject matter of the present invention as defined by Claims 1-6 (see In Re Bosy, 149 USPQ 792).

Claim 1 specifically claims a data structure in main memory. Webster's Seventh New Collegiate Dictionary, published by G. & C. Merriam Co., of Springfield, Massachusetts define a structure as:

(a) The elements of an entity or the position of such elements in their relationships to each other;

(b) The arrangements of particles or parts in a substance or body;

(c) Something made up of interdependent parts in a definite pattern of organization;

(d) Interrelation of parts as dominated by the general character of the whole.

The data structure system claimed is comprised of the first means which is the structure and arrangement of the running process word register at BAR+56 of the system base structure of Figure 6. The system base structure is described with the particularity of 35 USC 112 beginning on page 35 through 41. Its use is further described beginning on page 41. The running process word register structure is further shown with particu-

larity on Figure 11j and is described with particularity beginning on page 39. The second means which is an element within the first means is part of the running process word register and is identified with reference numeral 1129. The third means within the data structure is the J table word and the G table word at BAR and BAR+4 on Figure 6 and further shown on Figures 11a and 11b and described in the specification. Finally, the fourth means claimed is shown on Figures 13a and 13b is a central store as microprogrammed by the microprogram as disclosed by the flow charts of Figures 15b and 15c. Details of the control store which is microprogrammed is shown on Figure 13b (it should be understood that control stores are old in the art and persons of ordinary skill in the art can microprogram them when provided with the flow charts of the microprogram and this results in the firmware. It should be further noted that firmware should not be equated with the microprogram per se. The Computer Dictionary and Handbook, by Sippl et al. defines firmware as "logic circuits in read-only memory . . . . ."). Figures 13a and 13b are described with great particularity beginning on page 43 through page 53. Moreover, beginning on page 55 an example is taken by applying the microprogram flow chart of Figure 14a and actually microprogramming the control store of Figure 13b. Referring to Figure 14a and to page 55 of the specification the first block 1402 of Figure 14a instructs the microprogrammer to fetch the IPQW word from the system base into scratch pad memory. Beginning on line 16 of page 55 of the specification, there is shown the generation of the micro-operation signal 1360, 1361 to cause the retrieval of the IPQW word from the system base and storage subsystem 102, and transfer it to scratch pad memory 1315. Continuing further in the description of 14a there is similar descriptions with similar particularity to teach a person of ordinary skill in the art how to microprogram the control store or ROM in order to generate the firmware, first of the dispatch unit and more particularly the firmware of the instant invention. In In Re Chilowsky, 108 USPQ 321, the Court said "disclosure of application em-

braces not only what is expressly set forth in words or drawings, but what would be understood by a person skilled in the art." In In Re Karnofsky, 156 USPQ 682 the Court said in part "where one of ordinary skill in the art would know how to select operating conditions so as to achieve a particular result, the failure to include a recitation of some specific operating conditions and specifications cannot give rise to rejection either under 'enabling' or under 'best mode' requirement of 35 USC 112." Morevoer, In Re Karnofsky the Court said "Mere conclusory statements as to level of skill in the art are insufficient to controvert facts presented by Applicant as to sufficiency of specification's disclosure; it is not demonstrated how specification fails to meet the requirements of 35 USC 112." It is respectfully submitted that when the invention is reviewed in a factual manner as herein demonstrated by the Applicants that the specification discloses that which the Applicants consider as their invention with great particularity. A similar analysis may be conducted for the remaining claims.

The Examiner has further objected to the use of the term "basic data structure" as used on page 11 of the specification. Referring to page 11 the title under II is Basic Machine Structures. Continuing under the title the Applicants state "There are typically 3 basic data structures utilized in this hardware: data formats, software visible registers, and the instruction format." Webster's Seventh New Collegiate Dictionary previously referred to gives 2 definitions for "format" as a noun, are of which is as follows: "general plan of organization or arrangement." This is the definition intended by the Applicants and as such taken in context with the specification as a whole no confusion should result. In In Re Chilowsky, 108 USPQ 321 the Court said in part "disclosure of application embraces not only what is expressly set forth in words or drawings, but what would be understood by persons skilled in the art." In In Re Storrs, 114 USPQ 293 the Court in part "certainly required of disclosure by first paragraph of 35 USC 112 is not greater than that which is reasonable, having due regard to subject matter involved; in determining cer-

tainty required it cannot be forgotten that disclosure is not addressed to public generally, but to those skilled in the art." Along on the same subject matter in In Re Honn and Simms, 150 USPQ 652 the Court said in part "Applicants need not expressly set forth in specification that which would be understood by persons skilled in the art."

It is submitted that the Applicants have disclosed their invention in accordance with 35 USC 112 and the applicable case law. Applicants are further directing the Examiner's attention to other cases pertaining to disclosure of invention with requirements of 35 USC 112 as follows:

35 USC 112 CASES

(a) In Re Naquin (158 USPQ 317) 1968;
(b) In Re Ghiron and Ulrich (169 USPQ 793) 1971;
(c) In Re Brown (177 USPQ 691) 1973;
(d) In Re Knowlton (178 USPQ 486) 1973;
(e) In Re Comstock and Gilmer (178 USPQ 616) 1973;
(f) In Re Doyle (179 USPQ 237) 1973;
(g) In Re Brandstadter, Kiencle and Sykes (179 USPQ 286) 1973;

Moreover, the Examiner's attention is further directed to a book entitled, Microprogramming: Principles and Practices, by Samir S. Husson, published by Prentice-Hall Inc., of Englewood Cliffs, New Jersey in 1970. Moreover, the Examiner's attention is directed to an article entitled, Microprogramming Simplifies Control System Design, by Ron Kenny of Micro Data Corporation, Irvine, California. This article was published in Computer Design, the magazine of Digital Electronics, on February, 1975, Volume 14, No. 2. The address of the magazine is 221 Baker Avenue, Concord, Massachusetts.

In view of the above amendments, the discussion and the above authorities cited by the Applicants, the Examiner is respectfully requested to reconsider and remove his rejection based on an inadequate disclosure under 35 USC 112.

Claims 1-6 have further been rejected as drawn to subject matter which is inadequately disclosed under 35 USC 112 first paragraph. It is submitted that the amendment, discussions and authorities cited above also rebut this rejection and accordingly it is respectfully requested that the Examiner reconsider and remove this rejection.

Claims 1-6 have further been rejected under 35 USC 112, second paragraph, as failing to particularly point and distinctly claim the subject matter sought to be patented. The Examiner concludes in this rejection that "means refers to software of firmware instructions." Applicants are therefore requested to point out the structure in the drawings upon which the various means are to be read. This has been done with respect to the discussion supra. For example, Claim 1 as now amended was read in conjunction with the figures in the text of the specification. The first, second and third means are all hardware structure which store coded signals for performing a particular function as sanctioned under 35 USC 112 last paragraph. Software does not store signals. Moreover, the fourth means as now amended is also read upon the microprogrammed control store which generates signals which cause the particular function recited. Certainly this means is not claiming software. However, the fourth means is very definitely claiming firmware. It is not, however, claiming "firmware instructions" as the Examiner alleges. It has shown previously by quoting the Computer Dictionary and Handbook, by Sippl, et al. that firmware is "logic circuits in read-only memory . . ." It was previously shown how any control store having a ROM and more particularly the control store as disclosed in Figure 13a and 13b is microprogrammed with the flow charts of Figures 15b and 15c which then resulted in the first firmware claimed by the fourth means. Note the bare control store or ROM is not claimed and it is admitted is old in the art. Moreover, the flow charts of Figures 15b and 15c are not claimed per se. It is a combination of the microprogrammed control store which when microprogrammed in accordance with the microprogram depicted by flow charts of 15b and 15c results

in the firmware of the fourth means. The Examiner is further referred to the previously referenced book of Husson on "Microprogramming: Principles and Practices" which shows how persons of ordinary skill in the art talk to one another via flow charts in microprogramming control stores. It is submitted therefore that Claims 1-6 recite structure in the sense of apparatus, as clearly demonstrated herein. If the Examiner continues to maintain that this is not the case, he is respectfully requested to point out with particularity wherein the structural connections and cooperative relationships between various elements comprising the claimed combination have not been clearly established. In In Re Warner and Warner, 154 USPQ 174 the Court said "The Patent Office has initial duty of supplying *factual basis* for rejection; it may not, because it may doubt that invention is patentable, resort to speculation, unfounded assumptions or hindsight reconstruction to supply deficiencies in its factual basis, to extent that Office rulings are so supported, there is basis for resolving doubts against their correctness; likewise Court may not resolve doubts in favor of Office determination when there are deficiencies in record as to necessary factual bases supporting its legal conclusion or obviousness." (Underlining added.) Accordingly, in view of the above amendments to the claims, the arguments hereinbefore presented and the legal authorities cited, it is respectfully requested that the Examiner reconsider and remove his rejection under 35 U.S.C. 112 second paragraph.

The Examiner further rejected Claims 1-6 under 35 USC 101 as drawn to non-statutory subject matter. The Examiner said in making this rejection that "The Supreme Court in Gottschalk v. Benson, 175 USPQ 673, stated that Court's view on the patentability of certain claims which set forth an invention related to processing of data by a programmed computer.

"The Court in Benson addressed itself to the broad question of patentability of computer programs and stated that if such matter is to be patented, considerable

questions are raised which only committees of Congress can manage. Justice Douglas, speaking for the Court, declined to decide such policy matters and refused to extend the Patent Statutes to embrace computer programming, absent considered action by the Congress."

The Applicants traverse the Examiner's rejection of Claims 1-6 under 35 USC 101 as being drawn to nonstatutory subject matter. In the first instance it should be noted that the Benson application, upon which the Benson and Tabbot decision was based, included apparatus claims which were not the subject of the decision made by the Supreme Court. Apparently, it was considered that the apparatus claims did, in fact, define patentable subject matter while the process claims did not. However, the Benson invention was directed to a type of computer invention which happened to be implemented by means of a program.

As defined by Applicants' claims, the present invention is directed to hardware structures controlled by a microprogrammed control system which provides for the generation of signals which cause the creation of an image of a communication center at another location and for the generation of signals which causes the interchange of the information between the communication center and its image. (See Claim 2). It is difficult to understand how microprogram control systems can be said to be nonstatutory subject matter or in other words not pertaining to the technological arts. The technology or art to which Applicants' invention is directed is the same technology and art to which several of the patents cited by the Examiner are directed. Applicants submit that the extension of the holding in the Gottschalk v. Benson decision proposed by the Examiner would remove the majority of patents which have issued on computer program inventions (to which this invention does not pertain) from the purview of 35 USC 101.

Applicants submit that the structure and the microprogram control system defined by the Applicants' claims are clearly within the technological arts. The phrase "technological arts" is synonymous with the phrase" useful arts" as it appears in Article 1, Section 8 of the Con-

stitution (see In Re Waldbaum, 173 USPQ 430) and (In Re Johnston, 192 USPQ 272). As the Examiner may have already noted, the decision of Gottschalk v. Benson ruled that the invention as claimed was unpatentable because it was claimed too broadly and, therefore, "would wholly preempt the mathematical formula" and practically "would be a patent of the algorithm" (see In Re Christensen, 178 USPQ 35). Moreover, in its opinion in Dann v. Johnston which appears in Volume 188, No. 13 in a Supplement to the United States Patent Quarterly, Weekly Advance Sheets, dated March 29, 1976, the U.S. Supreme Court stated that in Benson, the respondent sought to patent as a "new and useful process" a method of programming a general purpose computer to convert signals from binary coded decimal form into pure binary form. The Court went on to say, "As we observe, the claims were not limited to any particular art or technology or to any particular apparatus or machinery or to any particular end use as quoted by the Examiner." With that, the Supreme Court finally stated in that regard that "Our limited holding was that the respondent's method was not a patentable 'process' as the term is defined in 35 USC 100(b)." Clearly, Applicants' claims are not concerned with a method of programming a general purpose digital computer to convert signals from binary coded decimal form into pure binary form. The Supreme Court has indicated that the Benson decision should be so limited.

More importantly, it should be noted that the U.S. Supreme Court in the Benson decision went so far as to issue broad exclusionary warnings prior to the final expression of unpatentability. For example, the Court said "it is said that the decision precludes the patent for any programming servicing a computer. We do not so hold." In a similar situation, Justice Frankfurter, in his concurring opinion given in the Supreme Court's decision of Funk Brothers Seed Company v. Kalo Innoculant Co., 76 USPQ 283, cited in the Gottschalk v. Benson decision, warned about the use of such terms as "works of nature" and "Laws of nature." He advised that these are such vague and malleable terms that the arguments drawn from such terms in ascertaining patentability

could fairly be employed to challenge almost every patent. Justice Frankfurter went on to say in that decision that, "Surely the Court does not mean to unwittingly pass on the patentability of such products by formulating criteria by which future issues of patentability may be prejudged.

From the above, it seems that by relying on the Gottschalk v. Benson decision to the extent of concluding that Applicants' claims are not limited to any particular art or technology, in accordance with the guidelines proposed in Gottschalk v. Benson, the Examiner is doing what Justice Frankfurter forewarned. Moreover, the Applicants submit that the Examiner's interpretation of the Benson decision and its application is contrary to what the Supreme Court intended for the reasons stated above.

In addition to the above statements, Applicants assume that by the term "program" the Examiner means its conventional meaning, as for example, one of the definitions set out in the text, in Computer Dictionary and Handbook, by Charles J. Sippl, et al., Copyright 1966 and 1972 by Howard W. Samms & Co., Inc. That is, a program is defined as one of the following definitions:

(1) A plan for the automatic solution of a problem; or,

(2) A set of instructions or steps that tells the computer exactly how to handle a complete problem . . . payroll, production scheduling or other applications; or,

(3) To plan a computation or process from the asking of a question to the delivery of results; or,

(4) To plan the method of attack for a defined problem.

Applicants submit that neither the claims nor specification is directed to a plan or set of instructions or steps, etc. Even, arguendo, if they were (which they are not) it is submitted that programs disclosed in accordance with 35 USC 112 are patentable subject matter under 35 USC 101. Several cases have spoken in this regard; and some of the leading ones are presented below:

CASES RE 35 USC 101

(a) In Re Abrams (89 USPQ 266) 1951;
(b) In Re Shao Wen Yuan (89 USPQ 324) 1951;
(c) In Re Jones (153 USPQ 77) 1967;
(d) In Re Tarczy-Hornoch (158 USPQ 141) 1968;
(e) In Re Prater and Wei (102 USPQ 541) 1969;
(f) In Re Bernhart and Fetter (163 USPQ 611) 1969;
(g) In Re Mahoney (164 USPQ 572) 1970;
(h) In Re Musgrave (167 USPQ 280) 1970;
(i) In Re Foster, et al (169 USPQ 99) 1971;
(j) In Re McIroy (170 USPQ 31) 1971;
(k) In Re Waldbaum (173 USPQ 430) 1972;
(l) In Re Christensen (178 USPQ 35) 1973.

Omitted from the above cases, all of which held that programs were patentable, subject matter is the Benson and Tabbot decision which as previously discussed supra is severely limited to its own fact situation and was decided by the Supreme Court on the basis that the subject matter was a mathematical algorithm which was claimed too broadly so as to preempt all future uses of the mathematical algorithm i.e. the case was a matter of overclaiming.

The holdings of the above cases are still good law and have not been overruled by Benson and Tabbot; as a matter of fact, Benson and Tabbot itself has been severely limited by the In Re Christensen decision and the In Re Johnston decision discussed supra. Apparently, the CCPA has been making law regarding computer programs, and the Supreme Court has seen fit not to disturb this law with the exception of Benson, wherein its decision was really on the ground of overclaiming so as to preempt all uses of the algorithm in a computer and not on the grounds of a pure program per se.

In view of the above discussion wherein it is pointed out that the subject invention is not a mathematical algorithm and is not directed to a program but rather to hardware and firmware and moreover, arguendo, even if it were it still would be patentable subject matter under

35 USC 101. Accordingly, it is respectfully requested that the Examiner reconsider and remove his rejection based on 35 USC 101.

Claims 1-6 have been further rejected under 35 USC 102 as unpatentable over the Hauck patent. The Examiner states "The apparatus in Figure 1 of Hauck cooperates to perform the functions of Applicants' claimed invention." The Examiner's rejection is hereby traversed and very strong exception is taken to the reason given for the Examiner's rejection on this basis. In the first place, the functions of the Hauck device are completely different than the functions of the instant invention. Moreover, the structure and hardware of the Hauck device are completely different than the structure of the instant invention. The Hauck patent discloses a push down stack in main memory wherein are stored "control words" which mark off procedural areas of the stack, operand words for processing and reference words for indicating locations of control words and of operand words. A pair of high speed registers are provided for coupling information to the processor's arithmetic unit for processing. These two registers are linked to the stacks so that they are the top two available locations of the stack. The processor includes a plurality of high speed registers which are linked to the memory stack to form a stack extension. Stored in the stack extension are control words that form a part of an address environment stack history list; parameters that are local to the current addressing environment; and operands that have been called to the top of the stack. Words stored in the stack extension are captured and are not pushed down into the memory stack during the execution of a procedure. Means are provided for addressing each register within the stack extension and for coupling captured information therein to the register which is the top available location of the stack. Means are provided for updating the stack extension when a new procedure is entered so that it will capture the information relating to the new procedure." (See Abstract of the Hauck patent).

Apparently, the Hauck patent discloses an improvement for a stack register by providing a stack extension to the stack register so that a desired item in the stack can be brought to the top of the stack without accessing main memory. See Columns 1-6 of the Hauck patent. An additional feature of the Hauck device is to store into main memory information related to the active stack that will not fit into the stack extension. Hauck states in column 6, lines 33-44 that "The stack extension according to the present invention which captures information, items and prevents them from being pushed down into main memory is important because the first items stored into the stack extension are usually parameters which are most often accessed for use in processing. Hence minimum access time to these parameters is achieved by keeping them in the higher speed stack extension. This is in contrast to the prior art stack extension where the first stored information items are pushed out of the bottom of the stack extension into main memory."

Stated in as few words as possible, it appears to the Applicants that the Hauck device pertains to an improvement of a stack type register wherein an item of information placed in the stack can be brought to the top of the stack or kept there via an extension stack without using indirect addressing techniques and the attendant multiplicity of main memory accesses.

The instant invention is quite different. The instant invention does not use a stack structure. The instant invention provides a communication area communicating between an operating system and a scratch pad memory which is located in the CPU.

This communication system is improved by the combination which causes an image of the communication center to be generated at another location in main memory, wherein the information in the image is then updated and then retransferred back to the communication center and to the scratch pad memory. This is an entirely different and novel operation, having entirely new, different and novel functions, and entirely new, novel and different combination of structures than the device of

Hauck and as a matter of fact any other devices of which the Applicants are aware. The claimed language of the instant invention reflects precisely these novel functions and novel combination of structures. (See Claim 2 in particular). If the Examiner persists in his rejection of Claims 1-6 under 35 USC 102, he is respectfully requested to point out with particularity how Figure 1 of the Hauck patent cooperates to perform the function of Applicant's claimed invention. In view of the amendments made to the claims and in view of the above discussion, it is respectfully requested that the Examiner reconsider and remove his rejection based on 35 USC 102.

References B-H which were not applied against the claims have been carefully reviewed but do not appear to add anything upon which the claims would read or be obvious in view of the reference.

In view of the amendments made herein and the above discussion, it is submitted that all the claims are now in an allowable form and allowance of Claims 1-6 is hereby respectfully requested and that the application be passed to issue at an early date.

If there are any questions which the Examiner feels can be resolved by a telephone call, he is respectfully requested to call the Applicants' Attorney collect at the telephone number listed below.

Respectfully submitted,

/s/ Nicholas Prasinos
Nicholas Prasinos
Registration No. 24,791
Attorney for Applicants

LETTER OF EXAMINER, OCTOBER 27, 1976.

☒ Responsive to communication filed on August 11, 1976.

☒ This action is made final.

A SHORTENED STATUTORY PERIOD FOR RESPONSE TO THIS ACTION IS SET TO EXPIRE THREE (3) MONTH(S) ~~DAYS~~ FROM THE DATE OF THIS LETTER.

FAILURE TO RESPOND WITHIN THE PERIOD FOR RESPONSE WILL CAUSE THE APPLICATION TO BECOME ABANDONED. 35 U.S.C. 133

PART I THE FOLLOWING ATTACHMENT(S) ARE PART OF THIS ACTION:

1. ☐ Notice of References Cited, Form PTO-892.
2. ☐ Notice of Informal Patent Drawing, PTO-948.
3. ☐ Notice of Informal Patent Application, Form PTO-152
4. ☐

PART II SUMMARY OF ACTION

1. ☒ Claims 1 - 6 are pending in the application.

   Of the above, claims _____ are withdrawn from consideration.

2. ☐ Claims _____ have been cancelled.
3. ☐ Claims _____ are allowed.
4. ☒ Claims 1 - 6 are rejected.
5. ☐ Claims _____ are objected to.
6. ☐ Claims _____ are subject to restriction or election requirement.
7. ☐ The formal drawings filed on _____ are acceptable.
8. ☐ The drawing correction request filed on _____ has been ☐ approved. ☐ disapproved.
9. ☐ Acknowledgement is made of the claim for priority under 35 U.S.C. 119. The certified copy has ☐ been received. ☐ not been received. ☐ been filed in parent application; serial no. _____ filed on _____.
10. ☐ Since this application appears to be in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under Ex parte Quayle, 1935 C.D. 11; 453 OG. 213.
11. ☐ Other

1) This is in response to the amendment filed August 11, 1976. Claims 1 through 6 remain in the application.

2) Claims 1 through 6 are again rejected as being directed to non-statutory subject matter under 35 USC 100 and 101.

Applicants contend that "the present invention is directed to hardware structures controlled by a microprogrammed control system which provides for the generation of signals which cause the creation of an image of a communication center at another location and for the generation of signals which causes the interchange of the information between the communication center and its image." Applicants further contend that "the structure and microprogram control system defined by applicants' claims are clearly within the technological arts" and is thus statutory subject matter within the intended bounds of 35 USC 100 and 35 USC 101.

Although applicants have now amended the claims to couch the claimed subject matter in language inferring structural entities and interconnections, the disclosed invention and specification language clearly indicate that the invention resides in a "technique" or "switch base instruction" for changing information in the system base in both the main memory and internal scratchpad registers making the system base in scratchpad memory accessible to software but not visible to software i.e. cannot be read by the software.

In claim 1, the preamble sets for the environment as a multiprogramming computer system comprised of such well known components as a main memory, central processing unit, and scratchpad registers. The subject matter deemed as the invention is then stated as "a data structure." The nature of this subject matter is disclosed beginning on page 35 of the specification and in Figure 6. Beginning in line 12 of page 35, it is stated that "the format of the system base 600 is shown." It is further stated that the "system base resides in absolute main memory and is developed by firmware." The term "data structure" as recited in the claim is comprised of four entities described as "first means", "second means",

"third means", and "fourth means." Whether these so called "means" are hardware structure which store coded signals for performing a particular function or whether they refer to coded signals representing computer control words per se, any life which the claim may have in view of that which is admittedly old in the art, resides in the "technique" (page 5, line 4) or "switch-system base instruction" (page 5, lines 13 and 14) for solving the problem of changing information of the system base which is located in scratchpad registers which software cannot access (page 4, line 20). The invention then is embodied in the coded signals representing instructions to the computer, the running process word RPW (first means of claim), priority level indicator word PRI (second means), J Table word and G Table word (third means), and micro-instructions of Figures 15c and 15c as found in the central store 1301. The invention resides in a "data structure" or an algorithm designed to control the multiprogramming computer to solve the particular problem indicated.

Under the ruling in *Gottschalk* v. *Benson*, 175 USPQ 673, the instant claims, depending upon a program implemented algorithm for patentability, are deemed nonstatutory subject matter.

3) All claims have been rejected. This action is hereby made a *FINAL REJECTION.*

CHARLES E. ATKINSON
EXAMINER
GROUP ART UNIT 236

LETTER TO OFFICE, DECEMBER 30, 1976.
AMENDMENT UNDER RULE 116

December 28, 1976

Commissioner of Patents & Trademarks
Washington
District of Columbia 20231

Dear Sir:

Responsive to the Office Action of October 27, 1976 and pursuant to Rule 116, please amend the above-identified application as follows:

REMARKS

Claims 1-6 are still active in the application.

Reconsideration of the instant application is respectfully requested.

Claims 1-6 have been rejected as being directed to non-statutory subject matter under 35 USC 100 and 101. The Examiner states in making this rejection as follows: "Although Applicants have now amended the claims to couch the claimed subject matter in language inferring structural entities and interconnections, the disclosed invention and specification language clearly indicates that the invention resides in a 'technique' or 'switch base instruction' for changing information in the system base both the main memory and internal scratchpad registers making the system base and scratchpad memory accessible to software but not visible to software i.e. cannot be read by the software. . . . The subject matter is deemed as the invention is then stated as 'a data structure'. . . . The term 'data structure' as recited in the claim is comprised of four entities described as 'first means', 'second means', 'third means', and 'fourth means'. Whether the so-called 'means' are hardware structure which store coded signals for performing a particular function or whether they refer to coded signals representing computer control words per se, any life which the claim may have in view of that which is immediately old in the art

resides in the 'technique' (page 5, line 4) or 'switch system base instruction' (page 5, lines 13 and 14) for solving the problem of changing information of the system base which is located in scratchpad registers which software cannot access (page 4, line 20)."

The Examiner then concludes in his discussion that "The invention then is embodied in the coded signals representing instructions to the computer, the running process word RPW (first means of claims), priority level indicator word PRI (second means), J table word and G table word (third means), and micro-instructions of Figures 15C and 15C(?) as found in the central store 1301. The invention resides in a 'data structure' or an algorithm designed to control the multiprogramming computer to solve the particular problem indicated."

Finally, the Examiner concludes "Under the ruling in Gottschalk v. Benson 175 USPQ 673, the instant claims, depending upon a program implemented algorithm for patentability, are deemed non-statutory subject matter."

It appears to the Applicants that the Examiner is using procrustean reasoning and forcing an arbitrary conclusion or solution to the facts. There appears to be a non-sequitur between the facts expounded and the conclusion reached. The claims are directed to data structures in hardware i.e. registers or computer memories. It can hardly be denied that registers and memories are hardware. The claims are directed to a specific new, novel and unobvious arrangement of these structures. This arrangement is more specifically recited in the claims and amplified in a discussion of Amendment A on pages 17-20 in this application.

More specifically, Claim 1 claims a data structure in main memory. Webster's Seventh New Collegiate Dictionary, published by G & C Merriam Company, Springfield, Massachusetts define a structure as:

(a) the elements of an entity or the position of such elements in their relationships to each other;

(b) the arrangements of particles or parts in a substance or body;

(c) something made up of interdependent parts in a definite pattern of organization;

(d) interrelation of parts is dominated by the general character of the whole.

The data structure system claimed is comprised of the first means which is the structure and arrangement of the running process word register at VAR+56 of the system base structure of Figure 6. The system base structure is described with the particularity of 35 USC 112 beginning on page 35 through 41. Its use is further described beginning on page 41. The running process word register structure is further shown with particularity on Figure 11J and is described with particularity beginning on page 39. The second means which is an element within the first means is part of the running process word register and is identified with reference numeral 1129. The third means within the data structure is the J table word and the G table word at BAR and BAR+4 on Figure 6 and further shown on Figures 11A and 11B and described in the specification. Finally, the fourth means claimed is shown in Figures 13A and 13B is a control store (i.e. ROM, etc.) as microprogrammed by the microprogram as disclosed by the flow charts of Figures 15B and 15C. Details of the control store which is microprogrammed is shown on Figure 13B (it should be understood that control stores are old in the art and persons of ordinary skill in the art can microprogram them when provided with a flow chart of the microprogram and this results in the firmware. It should be further noted that firmware should not be equated with microprograms per se. The Computer Dictionary and Handbook by Charles P. Sippl, et al, defines firmware as: "Logic circuits in read-only memory . . .)." Figures 13A and 13B are described with great particularity beginning on page 43 through page 53. Moreover, beginning on page 55 an example is taken by applying the microprogram flow chart of Figure 14A and actually microprogramming the control store of Figure 13B. Referring to Figure 14A and to page 55 of the specification, the first block 1402 of Figure 14A in-

structs the microprogrammer to fetch the IPQW word signals from the system base into scratchpad memory. Beginning on line 16 of page 55 of the specification, there is shown the generation of micro-operation signals 1360 and 1361 by hardware disclosed in accordance to 35 USC 112 to cause the retrieval of the IPQW word signals from the system base and storage subsystem 102 and transfer it to scratchpad memory 1350. Continuing further in the description of 14A there is similar descriptions with similar particularity to teach a person of ordinary skill in the art how to microprogram the control store or ROM in order to generate firmware, first of the dispatcher unit and more particularly the firmware of the instant invention. It should be noted that each and every means are hardware. The claimed language does not "*infer* structural entities and interconnections" as alleged by the Examiner, but actually claims structural entities and interconnections. The hardware handles and processes electrical signals, which is what computers do. The Examiner is incorrect when he states that the "disclosed invention and specification language clearly indicate that the invention resides in a 'technique' . . . ." The invention in any patent or patent application is what is claimed by the claim language. The specification may be used to interpret the bounds of the claim language. It appears that the Examiner is *clearly* disregarding the claim language and is urging upon us to interpret the invention strictly on the basis of "the specification language." This is contrary to accepted patent law. By reading the claim language, it will be seen that no "technique" is being claimed. Webster's Seventh New Collegiate Dictionary, by G & C Merriam Co., of Springfield, Massachusetts, published in 1969, define a technique as: "technical methods especially in scientific research"; "a method of accomplishing a desired aim." The Examiner is respectfully requested to reread the claim language and kindly inform the Applicants how hardware structure can arbitrarily be classified as a method or technique.

The Examiner also further alleges that "The disclosed invention and specification language clearly indicate that

the invention resides in 'a technique' or '*switch system base instruction*' for changing information in the system base in both the main memory and internal scratchpad registers making the system base in scratchpad memory accessible to software *but not visible to software i.e. cannot be read by the software.*" The apparent implication of this language is that instructions are techniques or methods and accordingly are unpatentable under 35 USC 100 and 101. It appears to the Applicants that the Examiner is confusing an algorithm of an instruction with the instruction itself. Instructions in computer programs have been patentable since time immemorial. It is well known in the computer art that what provides the power to a computer for accomplishing tasks are instructions. The more instructions a computer has, the greater power and functionality that it has to perform a variety of tasks. A machine capable of executing only two instructions is not as powerful as a machine capable of executing a hundred instructions; and it is obvious that these two machines are not the same machine. Instructions have been traditionally embodied in computers in three different ways as follows:

(a) by software programming i.e. (macroprogramming) of the computer (i.e. by placing the instruction capability temporarily in the computer).

(b) by microprogramming a control store (read only memory ROM) to generate firmware (i.e. by constructing the instruction capability to reside permanently in the computer; and,

(c) by building the instruction permanently in the computer in the form of hardware apparatus.

The first technique is the cheapest but also the slowest in operation; however, since it can be further programmed, alternatives can be made simply. The next technique, microprogramming, is a permanent embodiment in the control store (ROM memory) and is faster than the first technique but slightly more expensive. Finally, the last technique, building it into hardware, is the most expensive but it is very fast in execution. Today, almost

all major manufacturers utilize the second technique of embodying new functionality into a computer. The Applicants have disclosed the second technique (b)—i.e. firmware, as described supra. Firmware as previously noted supra has been defined by Sippl and Sippl as "logic circuits in read-only memory. . . ." Logic circuits are certainly patentable subject matter provided they meet the other requirements of patentability and there would be no question that logic circuits would not fall under 35 USC 100 or 101. Read only memories are also patentable subject matter and do not fall within 35 USC 100 or 101. Why then are novel logic circuits embodied in read only memories classified as nonpatentable subject matter? How can a piece of hardware be classified as a "technique" or a method?

It is submitted that the Examiner has erroneously interpreted the claims and classified the invention erroneously into a technique or a method. Even under this erroneous assumption, it would still not fall into the category of unpatentable subject matter because methods and techniques are patentable subject matter under 35 USC 100 and 101. It can only be inferred by the Applicants that what the Examiner was trying to establish is that the claimed subject matter is a mathematical algorithm and accordingly unpatentable under Gottschalk v. Benson & Tabbot. This, it is submitted, the Examiner has failed to do because in fact the invention is not a mathematical algorithm nor is it even an algorithm per se or even a method. Again the Applicants wish to stress it is hardware—that is what is disclosed and that is what is being claimed.

The Applicants would like to digress for a moment and submit that 35 USC 100 and 35 USC 101 were both included in the patent statutes to prevent individuals from trying to claim a scientific law, scientific principles, mathematical equations, pure mathematical algorithms, etc., which would preempt an entire scientific principle, mathematical formula, scientific law or method of doing business. It is obvious that merely because one discovers the law of gravity he should not be able to patent the law of gravity or that merely because one discovers relationships which pertain to basic fundamental laws

such as Einstein's $E=mc^2$ he should not be allowed to obtain a monopoly on the fundamental relationship or law itself via a patent. It is submitted, however, that inventions *based* on these scientific principles, mathematical equations, fundamental laws i.e. embodied into workable embodiments and utilizing such discoveries were never meant to be excluded by 35 USC 100 and 35 USC 101. (How many inventions have issued based on the law of gravity?) Yet this is what the Examiner in this instance is attempting to do—exclude a patentable invention merely because it is based on an algorithm. It is submitted that the Examiner is arbitrarily attempting to fit a hardware embodiment into a category in which it doesn't belong (i.e. a mathematical algorithm) so he can apply Benson & Tabbot and reject it on those grounds.

It is further submitted that at worst the Examiner has erroneously classified this invention as a *method* of operating a computer and at best that the invention *derives* from some general algorithm. Methods are patentable inventions provided they meet the criteria of patentability and should not categorically be classified as unpatentable subject matter. Moreover, inventions *derived* or *based* on scientific principles have been patentable since time immemorial. The majority or bulk of inventions that have been made since time immemorial derive from some scientific principles, basic scientific law, or mathematical algorithm. Surely the Examiner is not taking the position that all such inventions so derived are unpatentable. Moreover, merely because the scientific principle or theory on which the invention is based is described in the specification, does not negative the patentability of the embodiment of that scientific principle or theory. See In Re Storrs 114 USPQ 293 wherein it is stated "Applicant need not understand theory of operation of invention." See also In Re McCollough 136 USPQ 657 wherein it is stated "It is not customary in patent cases to require an Applicant either to know and disclose the underlying theory of his invention or to hold him to an expressed erroneous theory. Eames v. Andrews 122 US 40; Robinson on Patents, Paragraphs 485; In Re Modine 19 CCPA 1058, 57 Federal Second, 355, 13 USPQ 88. In principle,

the same would generally be true of an erroneous application o. theory, I think."

Accordingly, the Applicants should not be penalized merely because they disclose the scientific principle, algorithm, basic law or theory upon which their invention is based. Merely because the Applicants have gone the extra mile and disclosed the theory upon which their invention is based should not negative invention.

It is submitted therefore that the Applicants' invention is not, nor do the claims claim a theory, a mathematical algorithm, a scientific principle, or a fundamental physical law, and as a matter of fact, is not and does not claim a method of operating a computer. How then can the ruling in Gottschalk v. Benson 175 USPQ 673 apply? Several cases have spoken since Benson and Tabbot interpreting that case including In Re Christensen 178 USPQ 135, In Re Johnston 189 USPQ 257 and more recently the cases decided by the CCPA of In Re Noll and In Re Chatfield, all of which restrict the Benson and Tabbot decision to its own fact situation; that is the invention as claimed was unpatentable because it was claimed too broadly and, therefore, "would wholly preempt the mathematical formula" and practically "would be a patent on the algorithm itself." Judge Baldwin in In Re Noll said that claims define invention. Present claims (for In Re Noll) describe an apparatus, not a program. Claim must not be judged on a small abstracted portion deemed to be novel, but on the subject matter as a whole. The instant claims (in In Re Noll) are drawn to physical structure and not to an abstract "law of nature, a mathematical formula, or an algorithm." Appellants' machine is structurally different from a machine without a program. Moreover, instant claims (in In Re Noll) are limited to a particular technology, computer graphic systems and scan-conversion for graphic information. Furthermore, not all machines for achieving Appellants' results are included within the scope of Appellants' claims; non-program-controlled machines are clearly excluded. Court need not decide whether computer programs per se, constitute statutory subject matter. Moreover, Judge Baldwin further said that Benson does

not stand for a general proscription of patenting of computer programs and said that moreover, in review of Johnston, the Supreme Court emphasized that Benson was a limited holding.

In a nutshell (pardon the expression) then, Benson and Tabbot has been severely limited by subsequent interpretations of the CCPA. Benson and Tabbot stands for the proposition of claiming so broadly that the mathematical algorithm itself is claimed and thus there is preempted by the claim all uses of the mathematical algorithm. In the instant application it has been shown by a logical step by step progression that no mathematical algorithm is disclosed in the application nor claimed. As a matter of fact, no algorithm whatsoever is claimed. Moreover, no method or technique is claimed. What is claimed is strictly hardware; that whether or not this hardware embodiment claimed is developed from or based on a scientific principle, mathematical algorithm, law or nature, is immaterial as long as that mathematical algorithm, scientific principle or law of nature is not in itself claimed. It is submitted therefore that the reasoning of the Examiner in applying Benson and Tabbot is erroneous. It is erroneous both in fact and also in law. It is submitted that merely by calling something a mathematical algorithm, or a mathematical law, or a program does not make it so. Merely by placing a sign on a horse and saying "this is a cow" does not make that horse a cow, because the horse does not behave like a cow, and does not have the characteristics of a cow i.e. it does not have all the attributes of a cow, although it may have four legs, a tail, etc. To call hardware or firmware a program, or a mathematical algorithm, etc. does not make that firmware or hardware any more a program or mathematical algorithm because it does not have all the characteristics of a program or a mathematical algorithm. Accordingly, it is submitted that a procrustean solution does not stand up under the scrutiny of logic when attached either from the factual side or from the legal side. Accordingly, it is submitted that the instant invention as disclosed and claimed pertains to statutory subject matter under 35 USC 100 and 101 and

accordingly is allowable subject matter and allowance of Claims 1-6 is hereby respectfully requested and the application passed to issue at an early date.

If there is any way in which the application can be advanced to issue by a telephone call, the Examiner is requested to call the Applicants' Attorney collect.

Respectfully submitted,

/s/ Nicholas Prasinos
NICHOLAS PRASINOS
Attorney for Applicants
Registration No. 24,791

LETTER OF EXAMINER, FEBRUARY 1, 1977.

ADVISORY ACTION

[x] THE PERIOD FOR RESPONSE IS EXTENDED TO RUN 4 MONTHS FROM THE DATE OF THE FINAL REJECTION. 855 O. G. 1109.

[ ] Appellant's Brief is due in accordance with Rule 192 (a).

Applicant's response to the final rejection, filed Dec. 20, 1976, has been considered with the following effect, but it is not deemed to place the application in condition for allowance:

1. [ ] The proposed amendments to the claims and/or specification will not be entered and the final rejection stands because:

   a. [ ] There is no convincing showing under Rule 116 (b).
   b. [ ] They raise new issues that would require further consideration and/or search.
   c. [ ] They raise the issue of new matter.
   d. [ ] They are not deemed to place the application in better form for appeal by materially reducing or simplifying the issues for appeal.
   e. [ ] They present additional claims without cancelling a corresponding number of finally rejected claims.

2. [ ] Newly proposed or amended claims _____ would be allowed if submitted in a separately filed amendment cancelling the non-allowable claims.

3. [x] Upon the filing of an appeal, the proposed amendment [x] will be [ ] will not be, entered and the status of the claims in this application would be as follows:

   a. [ ] Claims _____ would be allowable.

   b. [x] Claims 1 - 6 would not be allowable.

   However: (See attachment for further explanation.)

   (1) [ ] The rejection of claims _____ on references is deemed to be overcome by applicant's response.

   (2) [ ] The rejection of claims _____ on non-reference grounds only is deemed to be overcome by applicant's response.

4. [ ] The affidavit, exhibit or request for reconsideration has been entered but does not overcome the rejection.

5. [ ] The affidavit or exhibit will not be admitted because applicant has not shown good and sufficient reasons why it was not earlier presented.

6. [ ] The application having been examined under the special accelerated examining procedure (M. P. E. P. 708.02), the proposed amendment has not been considered since it does not prima facie place the application in condition for allowance or in better condition for appeal.

Claims 1-6 are deemed properly rejected under 35 USC 101 as drawn to non-statutory subject matter. The U.S. Supreme Court in rendering a decision on the *Dann* v. *Johnston* case, 189 USPQ 257, held the invention to be obvious under 35 USC 103, thus reversing the CCPA. The Court did not render a decision on the 35 USC 101 issue and moreover did not modify its position taken in the *Benson* v. *Gottschalk* case.

In both the *In re Noll* and *In re Chatfield* cases the CCPA reversed the Board of Appeals decision. Until these decisions become final and since further review of clarification may be forthcoming, the PTO is continuing to examine computer program related applications in accordance with its current interpretation of the Supreme Court decision in the *Benson* case.

/s/ Charles E. Atkinson
CHARLES T. ATKINSON
EXAMINER
GROUP ART UNIT 236

BRIEF, APRIL 19, 1977.

JURISDICTION

This appeal is from a final rejection by the Primary Examiner of Group 236. The final rejection was dated October 27, 1976, and the Appellants then amended under Rule 116 on December 28, 1976, whereupon the Examiner gave an Advisory Action dated February 1, 1977 and extended the response period for 4 months from the date of the final rejection i.e. February 27, 1977. The Appellants filed a timely Notice of Appeal on February 21, 1977. Jurisdiction of this Board is invoked under the provisions of 35 USC Section 134 and CFR 1.191 of the Rules of Practice of the United States Patent Office in Patent Cases.

STATUS OF THE CLAIMS

Claims 1-6 are pending. Each of these six claims has been finally rejected and such rejection has been appealed.

CONCISE EXPLANATION OF THE INVENTION

This invention generally concerns a data processing system provided with a plurality of peripherals and having multiprocessing and multiprogramming capabilities. More specifically, the invention concerns a combination of hardware including a model-independent instruction means which is implemented in firmware, for adding capability and increased functionality to the computer system. As defined by the Applicants' claims, the present invention is directed to a combination of hardware structures controlled by a microprogrammed control system which provides for the generation of signals which cause the creation of an image of a communication center at another location, and for the generation of signals which causes the interchange of the information between the communication center and its image.

The computer system shown generally in Figure 1 is comprised of a main memory storage subsystem 102, a

processor subsystem 101, a peripheral subsystem 103. As shown on Figure 12, a portion of the main memory hardware is reserved for hardware use and system software use. A system base 1215 is defined as a communication center between the operating system and the various processes that may execute on the computer system. This is done in general by providing a BAR (Boundary Address Register) 1202 to address various entities in the system base 1215 which then can address by indirection any number of process control blocks, PCB's 1206, each processing control block being associated with a particular process that can execute in the system. Accordingly, the computer hardware (i.e. registers, scratchpad memory 1315, peripherals, etc.) and the operating system (not shown, but described in reference 12, page 3 of the specification and incorporated by reference therein) can communicate directly with any of myriads of processes that can execute on the computer, without programmer intervention. The central processing unit CPU is comprised generally as shown on Figure 13a of an arithmetic logic unit ALU 1317, the instruction fetch unit IFU 1318, the address control unit ACU 1319, several general, base and scientific registers and scratchpad memory. The control unit including formats of microwords and the logic hardware for generating control signals in response to the microwords and microprogramming, is shown in Figure 13b. Associated with the ALU 1317 and the CSU 1301, is a scratchpad memory unit LSU 1315 (sometimes referred to also as the local store unit). The operations of the arithmetic logic unit are completely dependent on micro-operation control signals from the control store unit shown as CSU 1301 and Figure 13b. Scratchpad memory 1315 is used to store CPU control information and maintainability information. In addition, the scratchpad memory 1315 contains working locations which are primarily used for temporary storage of operands and partial results during the data manipulation.

With respect to this invention, the system base, which is located at a location specified by the boundary address register BAR in main memory, is copied into a new loca-

tion in main memory with predetermined portions of its information changed. This is readily accomplished by the operating system. (See application reference 12, on page 3, of the specification incorporated by reference into the application herein). The switch system base instruction then transfers the information which is in the system base at the new location into the system base at the location specified by BAR and also that portion of the system located in scratchpad memory. As defined by Applicants' claims, this invention is directed to hardware structures controlled by a microprogrammed control system Figure 13b, which provides for the generation of signals which cause the creation of an image of a communication center at another location and for the generation of signals which causes the interchange of the information between the communication center and its image. (See Claim 2). This communication system is provided by the claimed combination which causes an image of the communication center to be generated at another location in main memory, wherein the information in the image is then updated and then retransferred back to the communication center and to the scratchpad memory.

Claim 1 claims the means for storing coded signals of addresses and priority to address any selected one of the processes, for generating signals which cause the exchange of information between the claimed data structures and said scratchpad registers.

Claim 2 claims means for generating signals which cause the creation of an image in the communication center and claims means for generating signals causing an exchange of information within the created image and further claims means for generating signals which cause a transfer of the new image to the communication center and to the scratchpad memory.

Claim 3 further limits and defines the third means of Claim 2.

Claim 4 includes in the claimed combination the arithmetic and logic unit having an auxiliary memory.

Claim 5 claims additional hardware—the hardware gate.

Claim 6 further limits Claim 5.

* * * *

For Claims 1 to 6 inclusive, see Rejected Claims 1 to 6 inclusive.

REJECTION

Claims 1-6 have been rejected finally under 35 USC 101 as drawn to non-statutory subject matter. The Examiner states "The U.S. Supreme Court in rendering a decision on the Dann v. Johnston, 189 USPQ 257, held the invention to be obvious under 35 USC 103, thus reversing the CCPA. The Court did not render a decision on the 35 USC 101 issue and moreover did not modify its position taken in the Benson v. Gottschalk case.

In both the In Re Knoll and In Re Chatfield cases the CCPA reversed the Board of Appeals decision. Until these decisions become final and since further review of clarification may be forthcoming, the PTO is continuing to examine computer programs related application in accordance with its current interpretation of the Supreme Court decision in the Benson case."

ARGUMENT

It appears to the Appellant that the Examiner is using procrustean reasoning and forcing an arbitrary conclusion or solution to the facts. There appears to be a non-sequitur between the facts expounded and the conclusion reached. The claims are directed to a combination of hardware and firmware i.e. registers or computer memories etc. It can hardly be denied that registers and memories are hardware. The claims are directed to a specific new, novel and unobvious arrangement of these structures. This arrangement is more specifically recited in the claims and amplified in a discussion on pages 17-20 in this application.

Claim 1 specifically claims a data structure in main memory. Webster's Seventh New Collegiate Dictionary, published by G & C Merriam Company of Springfield, Massachusetts, define a structure as:

(a) The elements of an entity of a position of such elements in their relationship to each other;

(b) The arrangements of particles or parts in a substance or body;

(c) Something made up of interdependent parts in a definite pattern of organization;

(d) Interrelationships of parts as dominated by the general character of the whole.

A data structure claimed is comprised of the first means which is the structure of the running process word register at BAR + 56 of the system base structure of Figure 6. The system base structure is described with the particularity of 35 USC 112 beginning on page 35 through 41. Its use is further described beginning on page 41. The running process word register structure is further shown with particularity on Figure 11j and is described with particularity beginning on page 39 of the specification. The second means which is an element within the first is part of the running process word register and is identified with reference to numeral 1129. The third means within the data structure is the J table word means and the G table word means at BAR and BAR + 4 on Figures 11a and 11b and described in the specification. Finally, the fourth means claimed is shown on Figure 13a and 13b is a central store as microprogrammed by the microprogram as disclosed by the flow charts of Figures 15b and 15c. Details of the control store which is microprogrammed are shown on Figure 13b (it should be understood that control stores are old in the art and persons of ordinary skill in the art can microprogram them when provided with the flow charts of the microprogram and this results in the firmware. It should be further noted that firmware should not be equated with the microprogram per se. The Computer Dictionary and Handbook, by Sippl and Sippl, defines firmware as "logic circuits in read-only memory . . .)." Figures 13a and 13b are described with great particularity beginning on page 43 through 53 of the specifica-

tion. Moreover, beginning on page 55, an example is taken by applying the microprogram flow chart of Figure 14a and actually microprogramming the control store of Figure 13b to generate the signals for controlling certain predetermined tasks and functions. For example, referring to Figures 13b and 14a and to page 55 of the specification, the first block 1402 of Figure 14a instructs the hardware of Figure 13b to generate signals to fetch the IPQW word from the system base into scratchpad memory. (See sample generation of such signals by the hardware of Figure 13b). Beginning on line 16 of page 55 of the specification, there is shown the generation of the micro-operation signal 1360, 1361 to cause the retrieval of the IPQW word from the system base and storage subsystem 102, and transfer it to scratchpad memory 1315. Continuing further in the description of 14a, there is similar descriptions with similar particularity to teach a person of ordinary skill in the art how to microprogram the control or ROM in order to generate the firmware, first of the dispatch unit and more particularly the firmware of the instant invention. It should be noted that every means claimed is hardware. The claimed language does not "infer structural entities and interconnections" as alleged by the Examiner, but actually claims structural entities and interconnections. The hardware handles and processes electrical signals, which is what computers do. The Examiner is incorrect when he states that the "disclosed invention and specification language clearly indicates that the invention resides in a 'technique'." The invention in any patent or patent application is what is claimed by the claimed language. The specification may be used to interrupt the bounds of the claimed language. It appears that the Examiner is clearly disregarding the claimed language and is urging upon us to interpret the invention strictly on the basis of his interpretation of the specification language." This is contrary to accepted patent law. By reading the claimed language, it will be seen that no "technique" is being claimed. Webster's Seventh New Collegiate Dictionary, by G & C Merriam Co. of Springfield, Massachusetts, published in 1969, de-

fine a technique as "technical methods especially in scientific research"; "a method of accomplishing a desired aim." The Examiner is clearly wrong in arbitrarily classifying hardware structure as a method or technique.

The Examiner also further alleges that "the disclosed invention and specification language clearly indicates that the invention resides in 'a technique' or 'switch system base instruction' for changing information in the system base in both the main memory and internal scratchpad registers. Changing the system base in scratchpad memory accessible to software but not visible to software i.e. cannot be read by the software." The apparent implications of this language is that instructions are techniques or methods and accordingly are unpatentable under 35 USC 100 and 101. It appears to the Applicants that the Examiner is confusing an algorithm of an instruction with an instruction itself embodied in some medium. Instructions embodied in computer systems have been patentable since time immemorial. It is well known in the computer art that what provides the power to a computer for accomplishing tasks are instructions. The more instructions a computer has, the greater power and functionality that it has to perform a variety of tasks. A machine capable of executing only two instructions is not as powerful as a machine capable of executing 100 instructions; and it is obvious that these two machines are not the same machine. Instructions have been traditionally embodied in computers in three different ways as follows:

(a) By software programming i.e. (macroprogramming of the computer), i.e. by placing the instruction capability temporarily in the computer (i.e. in evanescent main memory form).

(b) By microprogramming a control store (read only memory ROM) to generate firmware (i.e. by constructing the instruction capability to reside permanently in the computer (i.e. logic circuits in ROM memory, generally in the form of integrated circuits); and,

(c) By building the instruction permanently in the computer in the form of hardware apparatus (i.e. discrete logic circuits).

The first technique is the cheapest but also the slowest in operation; however, since this can be further programmed and easily changed, alterations can be made simply. The next technique, microprogramming, is a permanent embodiment in the control store (ROM) memory and is faster than the first technique but slightly more expensive.

Finally, the last technique, building it into hardware, is the most expensive but it is very fast in execution. Today, almost all major manufacturers utilize the second technique (b) of embodying new functionality into computers. The Applicants have disclosed the second technique (b)—i.e. firmware, as described supra. Firmware as previously noted supra has been defined by Sippl and Sippl in their authoritative Computer Dictionary and Handbook as "logic circuits in read only memory . . . ." Large scale integration (LSI) of circuits have made this possible. Logic circuits are certainly patentable subject matter provided they meet the other requirements of patentability and there would be no question that logic circuits would not fall under 35 USC 100 or 101. Read only memories are also patentable subject matter and do not fall within 35 USC 100 or 101. Why then are novel logic circuits embodied in read only memories classified as non-patentable subject matter? How can a piece of hardware be classified as a "technique" or a "method"?

It is submitted that the Examiner has erroneously interpreted the claims and classified the invention erroneously into a technique or a method. Even under this erroneous assumption, it would still not fall into the category of unpatentable subject matter under 35 USC 100 and 101 because methods and techniques are patentable subject matter under 35 USC 100 and 101. See the following cases:

(a) In Re Abrams (89 USPQ 266) 1951;
(b) In Re Shao Wen Yua (89 USPQ 324) 1951;

(c) In Re Jones (153 USPQ 77) 1967;
(d) In Re Tarczy-Hornoch (158 USPQ 141) 1968;
(e) In Re Praeter and Wei (102 USPQ 541) 1969;
(f) In Re Bernhart and Fetter (163 USPQ 611) 1969;
(g) In Re Mahoney (164 USPQ 572) 1970;
(h) In Re Musgrave (167 USPQ 280) 1970;
(i) In Re Foster, et al (169 USPQ 99) 1971;
(j) In Re McIlroy (170 USPQ 430) 1972;
(k) In Re Waldbaum (173 USPQ 430) 1972;
(l) In Re Christensen (178 USPQ 35) 1973.

Omitted from the above cases, all of which held that programs were patentable subject matter, is the Benson and Tabbot decision since subsequent decisions of the Supreme Court (i.e. In Re Johnston 183 USPQ 172) and the CCPA (i.e. In Re Christensen 178 USPQ 35, and in In Re Noll 191 USPQ 271), and In Re Chatfield 191 USPQ 73), have seriously limited the Benson & Tabbot decision to its own fact situation of overclaiming so that it "would wholly preempt the mathematical formula", and particularly "would be a patent on the algorithm itself."

It can only be inferred by the Applicants that what the Examiner was trying to establish is that the claimed subject matter is a mathematical algorithm and accordingly unpatentable under Gottschalk v. Benson and Tabbot. This, it is submitted, the Examiner has failed to do because in fact, the invention is not a mathematical algorithm, nor is it even an algorithm per se, or even a method. Again, the Applicants wish to stress it is a combination of hardware and firmware—that is what is disclosed in the specification and that is what is being claimed.

The Applicants would like to digress for a moment and submit that 35 USC 100 and 35 USC 101 were both included in the patent statutes to prevent individuals from trying to claim a scientific law, scientific principles, mathematical equations, pure mathematical algorithms, methods of doing business, etc., which would preempt an entire scientific principle, mathematical formula, scien-

tific law or method of doing business. It is obvious that merely because one discovers the law of gravity, he should not be able to patent the law of gravity or that merely because one discovers relationships which pertain to basic fundamental laws such as Einstein's $E = mc^2$, he should not be allowed to obtain a monopoly on the fundamental relationships alone or law itself via a patent. It is submitted, however, that inventions *based* on these scientific principles, mathematical equations, fundamental laws, i.e. embodied into workable embodiments and utilizing such discoveries were never meant to be excluded by 35 USC 100 and 35 USC 101. (How many inventions have issues based on the law of gravity)? Yet this is what the Examiner in this instance is attempting to do—exclude a patentable invention merely because it is based on an algorithm. It is submitted that the Examiner is arbitrarily attempting to fit a hardware embodiment into a category in which it doesn't belong (i.e. a mathematical algorithm) so he can apply Benson & Tabbot and reject it on those grounds. If this were permitted, then hardly any computer related invention would be patentable since most computer system inventions are based on some form of an algorithm. As noted before, this algorithm may be implemented in software, firmware or hardware. It is submitted that Benson & Tabbot was finally rejected because the invention claimed would preempt a mathematical algorithm and would be a monopoly on the algorithm itself. Nor was the claimed invention in Benson & Tabbot restricted with respect to application or field of endeavor.

It is further submitted that at worst the Examiner has erroneously classified this invention as a *method* of operating a computer and at best that the invention *derived* from some general algorithm. Methods are patentable inventions provided they meet the criteria of patentability and they should not be categorically classified as unpatentable subject matter. Moreover, inventions *derived* or *based* on a scientific principle, have been patentable since time immemorial. The majority or bulk of inventions that have been made since time immemorial derive from some scientific principle, basic scientific law, or

mathematical algorithm. Surely the Examiner is not taking the position that all such inventions so derived are unpatentable. Moreover, merely because the scientific principle or theory of which the invention is based is described in the specification, does not negative the patentability of the embodiment of that scientific principle or theory to a hardware invention. See In Re Storrs 114 USPQ 293 wherein it is stated: "Applicant need not understand theory of operation of invention." See also In Re McCullough 136 USPQ 657 wherein it is stated: "It is not customary in patent cases to require an Applicant either to know and disclose the underlying theory of his invention, or to hold him to an express erroneous theory. Eames v. Andrews 122 US 40; Robinson on Patents, Paragraph 485; In Re Modine 19 CCPA 1058; 57 Federal Second, 355, 13 USPQ 88. In principle, the same would generally be true of an erroneous application of theory I think." Accordingly, the Applicants should not be penalized merely because they disclose the scientific principle, algorithm, basic law or theory upon which their invention is based. Merely because the Applicants have gone the extra mile and disclosed the theory upon which their invention is based, should not negative invention.

It is submitted therefore that the Applicants' invention is not, nor do the claims claim a theory, a mathematical algorithm, a scientific principle, or a fundamental physical law, and as a matter of fact, is not and does not claim a method of operating a computer. How then can the decision in Gottschalk v. Benson 155 USPQ 673 apply in the instant invention? Several cases have spoken since Benson & Tabbot interpreting that case including In Re Christensen 158 USPQ 135, In Re Johnston 189 USPQ 257 and more recently the cases decided by the CCPA, In Re Noll and In Re Chatfield (citations supra), all of which restrict the Benson & Tabbot decision to its own fact situation; that is the invention as claimed in Benson & Tabbot was unpatentable because it was claimed too broadly, and therefore, "would wholly *preempt the mathematical formula*" and practically "would be a *patent on the algorithm itself*." Judge Baldwin in In Re Noll said that claims define invention. The present claims (for

In Re Noll) describe an apparatus, not a program. Claims must not be judged on a small abstracted portion deemed to be novel, but on the subject matter as a whole. The instant claims (In Re Noll) are drawn to physical structure and not to an abstract "law of nature, a mathematical formula, or an algorithm." Appellants' machine is structurally different from a machine without a program. Moreover, instant claims (in In Re Noll) are limited to a particular technology, computer graphic systems and scan-conversion for graphic information. Furthermore, not all machines for achieving Appellants' results are included within the scope of Appellants' claims; non-programmed controlled machines are clearly excluded. The Courts need not decide whether computer programs per se, constitute statutory subject. Moreover, Judge Baldwin further said that Benson does not stand for a general proscription of patenting of computer programs and said that moreover, in review of Johnston, the Supreme Court emphasized that Benson was a limited holding.

In a nutshell, (pardon the expression) then, Benson & Tabbot has been severely limited by subsequent interpretations of the CCPA and by its own decision in In Re Johnston, Benson & Tabbot stand for the proposition of claiming so broadly that the mathematical algorithm itself is claimed and thus there is preempted by the claims all uses of the *mathematical algorithm.* In the instant application it has been shown by a logical step by step progression that no mathematical algorithm is disclosed in the application nor claimed. As a matter of fact, no algorithm whatsoever is claimed. Moreover, no method or technique is claimed. What is claimed is a combination comprised of strictly hardware; that whether or not this hardware embodiment claimed is developed from or based on a scientific principle, mathematical algorithm, law of nature is immaterial as long as that mathematical algorithm, scientific principle or law of nature is not *in itself* claimed.

Nor is a program claimed by the instant invention. Moreover, it is submitted that the claimed language is not a subterfuge for patenting a program. Note that In

Re Noll and In Re Chatfield, the subject matter was acknowledged to be a program—not firmware, and not hardware. It is submitted that the instant invention has been attacked merely because of guilt by association—by the company it keeps—in that it is computer related, and takes advantage of the latest advances in the computer art for its embodiment. This statement is given added weight by the Examiner's statement in his Advisory Action of 2/1/77 wherein he states:

"In both the In Re Noll and In Re Chatfield cases, the CCPA reversed the Board of Appeals' decision. Until these decisions become final and since further review of clarification may be forthcoming, the PTO is continuing to examine *computer program related applications* in accordance with its current interpretations of the Supreme Court decisions in the Benson case." (Underlining added).

The theory of guilt by association has long been discredited. It should not be revived now in one of the most progressive agencies of the U.S. Government by procrustean reasoning. It is submitted, therefore, that the reasoning of the Examiner in applying Benson & Tabbot is erroneous. It is erroneous both in fact and also in law. It is submitted that merely by calling something a mathematical algorithm or a mathematical law, or a program does not make it so. Merely by placing a sign on a horse and saying "this is a cow" does not make that horse a cow, because the horse does not behave like a cow, and does not have the characteristics of a cow, i.e. does not have all the attributes of a cow, although it may have four legs, a tail, etc., and in many ways may be similar to a cow. To call hardware or firmware a program, mathematical algorithm, etc., does not make that firmware or hardware any more a program or mathematical algorithm because it does not have all the characteristics of a program or a mathematical algorithm. Accordingly, it is submitted that the proscrustean solution of the Examiner does not stand up under the scrutiny of logic when attacked either from the factual side or from the legal side. Accordingly, it is submitted that the instant invention as disclosed and claimed pertains to statu-

tory subject matter under 35 USC 100 and 101 and accordingly is allowable subject matter.

CONCLUSION

Appellants respectfully submit that the claims in issue have been shown to be clearly allowable subject matter which is and always has been patentable under 35 USC 100 and 35 USC 101. Therefore, Appellants respectfully request that the Examiner's rejection of the appealed claims be reversed and that the application be allowed.

ORAL HEARING ON REQUEST

Appellants hereby request an Oral Hearing. Two extra copies of this Brief are provided in accordance with Rule 192.

Respectfully submitted,

/s/ Nicholas Prasinos
NICHOLAS PRASINOS
Attorney for Appellants
Registration No. 24,791

EXAMINER'S ANSWER, AUGUST 4, 1977.

This is an appeal from the final rejection of claims 1 through 6, all the claims remaining in the application.

A correct copy of the claims appear on pages 6 through 8 of applicants' brief.

STATEMENT OF GROUND OF REJECTION:

Claims 1 through 6 are rejected under 35 U.S.C. 101 as drawn to non-statutory subject matter under the ruling of the Supreme Court decision, Gottschalk v. Benson, 409 US 63, 175 USPQ 673.

In a subsequent Court decision, *In re Johnston*, 582 F.2d 765, 183 USPQ 172 (CCPA 1974), the Court held the claims unpatentable under 35 USC and did not render a decision on the 35 USC 101 rejection. Moreover, the Court did not modify its position taken in the *Benson* case.

On November 18, 1976, the Court of Claims and Patent Appeals decided the cases of *In re Noll* and *In re Chatfield*. In both cases the Board of Appeals had taken the position that the claims were drawn to non-statutory subject matter under the *Benson* decision. The CCPA reversed the Board of Appeals in both cases. However, per O.G. Notice 954 O.G. 550 dated January 11, 1977, until these decisions become final and since Request for Certiorari has been filed in both cases, the patent examining corps is directed to continue to examine computer programming applications in accordance with its current interpretation of the Supreme Court decision in the *Benson* case.

Applicant has set forth the background and general field of the invention beginning with page 3 of the application. The general field of invention is stated as that of a system and method for communicating between hardware and software in a multiprogramming/multiprocessing computer environment. It is then stated that multiprogramming/multiprocessing computer systems re-

quire the use of a communications area in main storage for communications between hardware and software. Applicants further state that high performance computing systems find it necessary to store portions of the system base in internal scratchpad registers instead of main memory and consequently the portion of the system base in the scratchpad registers is inaccessible and invisible to software running in the system. The problem to which this invention is directed was stated by applicants as the condition that arises when "software (i.e. the programmer) wants to change positions or change information of the system base which happens to be located in these internal central processor scratchpad registers which software cannot access". Applicants proceed to discuss two prior art methods or techniques to solve the problem and states their disadvantages and shortcomings. On page 5 of the specification, the objectives of the instant invention are set forth and include such statements as "a primary object of the invention is to provide an improved system and method for changing information in the system based of a computer system", "another object of the invention is to provide a switch-system base instruction in firmware which changes, upon execution, information in the system base". Thus, applicants have established the field of the invention, the problem to be solved, and have further given an indication of that which is deemed old in the art and that which is to be deemed their contribution to the art. Indeed, the claims define the invention but the claims must be interpreted in light of the disclosure.

In the amendment filed August 11, 1976, an Appendix I is set forth to provide further background of the invention and is specifically addressed to the first Office Action rejection wherein all the claims were rejected as drawn to inadequately disclosed subject matter. In this appendix, applicants set forth publications and U.S. patents as documented evidence that certain systems and devices required to carry out the invention in the instant application are well known in the data processing art. The examiner agrees that these publications and patents do point out the fact that the operating system within which

applicants' invention is utilized is well known in the art. There can be no controversy it would appear that the prior art does comprise multiprogramming/multiprocessing computer system comprising a CPU, main memory scratchpad registers, etc. and means for communicating between the operating system and the processes.

Applicants have stated in the brief, page 9, that "claim 1 specifically claims a data structure in main memory". It would appear then that applicants and the examiner agree as to the nature of applicants' contribution to the art and that the portion of claim 1 which sets forth the subject matter deemed to distinguish over that which is old in the art is the portion defining the "data structure". Applicants further state in the brief, page 13, that they have chosen to implement their improvement of an existing operating computer system by the second of three "techniques". This second technique of implementing "instructions" into a computer is generally categorized as "firmware" as opposed to software programming and hardwired apparatus. Applicants contend that their "firmware" invention when loaded into the registers of the operating system result in an invention having structural aspects and hardware limitations. Claim 1 has been couched in apparatus format and recites first means through fourth means. The first means through third means set forth means for storing signals and read upon registers or memory locations within the operating system, already shown to be old in the art. The function of or attribute of the signals stored in the first, second, and third means for storing may be classified as indicating an address for a selected one of a plurality of processes, indicating a priority of said selected process relative to other processes, and indicating an address of a selected one of the plurality of processes, respectively. There is clearly no novelty to be found here for distinguishing over that which is deemed to be old in the art.

The fourth means is stated as means for generating signals causing the changing of information in the data structure and scratchpad registers and is categorized by applicants' brief, page 10, as a control store as micropro-

grammed by the microprogram of Figures 15b and 15c. The use of control stores in computer systems has been shown to be old in the art in the listing of publications and U. S. Patents in paragraph 19 of Appendix I, amendment filed August 11, 1976. In fact, paragraph 22 of the Appendix I correctly indicates that microprogrammed computer processors including microprogrammed control units are old in the art and suggests that the systems shown in the two cited patents are capable of practicing or implementing the instant invention.

Applicants' contention that *Benson* is not properly applied against the claimed invention because of the hardware limitations expressed in the claims cannot stand. As correctly stated in the objectives of the invention within the disclosure and as pointed out in the previous discussion of the claim recitation in light of that which is deemed old in the art, the invention resides in a computer instruction in firmware. The so called limitations expressed in claim 1 are so conventional that it would have been the exceptional computer at the time of applicants' filing date which would not include such limitations. Applicants' invention resides in an instruction in firmware which changes, upon execution, information in the system base. The claim is thus reciting prior art coupled with subject matter which the U.S. Supreme Court has found to be non-statutory in *Benson*. As stated by Justice Douglas in *Benson*:

> "It may be that the patent laws should be extended to cover these programs, a policy matter to which we are not competent to speak.
>
> * * * *
>
> "If these programs are to be patentable, considerable problems are raised which only committees of Congress can manage, for broad powers of investigation are needed, including hearings which canvass the wide variety of views which those operating in this field entertain. The technological problems tendered red in the many briefs before us indicate to us that considered action by the Congress is needed".

Whereas the previous discussion has been limited to claim 1, the same rationale applies equally to claim 2. The recitation of claim 2 perhaps more clearly points out the contribution of applicants to the prior art. The three means recited as comprising the improvement are recited as means for generating signals to cause the computer to perform certain functions. These means must be construed to read upon memory locations comprising instructions required to command and control the computer elements for performing the functions recited. Applicants' contribution to the art resides in the specific instructions found within the memory locations and not the combination of three memory locations labeled first, second, and third means.

The examiner must take exception to the statement on page 14 of the brief wherein the list of cases labeled (a) through (b) are referred to as holding that programs are patentable subject matter. The examiner has read most, if not all, of the above cases at some point in time and must disagree vehemently with applicants' observation as to their holding that computer programs are statutory subject matter under 35 U.S.C. 101. The examiner's observation would be that none of the listed cases hold that computer programs are statutory subject matter.

The examiner respectfully submits that claims 1 through 6 are properly rejected as drawn to non-statutory subject matter under 35 U.S.C. 101.

/s/ Charles E. Atkinson
CHARLES E. ATKINSON
EXAMINER
GROUP ART UNIT 236

CONFEREE: M. A. MORRISON

06/28/77

REPLY BRIEF, AUGUST 17, 1977.

GENERAL

This Reply Brief is necessitated because the Examiner has raised new points and new issues in his answer of August 4, 1977. Under 37 CFR 1.193(b) the Appellant has a right to file a Reply Brief within twenty days from the date of Examiner's answer directed to such new points or arguments as have been raised by the Examiner's answer.

NEW POINTS AND ARGUMENTS RAISED BY THE EXAMINER

The Examiner's answer of August 4, 1977 attacks the elements of the Appellants' claims as being old in the art without citing references and without applying anticipation or obviousness rejections; accordingly no weight should be given to such arguments. Since the Examiner has not applied either a 35 USC 102 or a 35 USC 103 rejection to the claims, it is respectfully requested that all references to elements that are old in the art, and by inference anticipated or obvious, be stricken from the record, since neither anticipation or obviousness under 35 USC 102 or 35 USC 103 is an issue in the instant appeal. Examples of such references by the Examiner to elements being old in the art are as follows:

1. Page 5, beginning on line 6, it is stated "Claim 1 has been couched in apparatus format and recites first means through fourth means. The first means through third means set forth means for storing signals and read upon registers or memory locations within the operating system, already shown to be all old in the art . . . there is clearly no novelty to be found here by distinguishing over that which is deemed to be old in the art."

2. On page 5, beginning on the 6th line from the bottom, it is stated "The fourth means is stated as means for generating signals causing the changing of information in the data structure and scratchpad registers and is categorized by Applicants' Brief, page 10, as a control store as microprogrammed by the microprogram of Fig-

ures 15b and 15c. The use of control stores in computer systems has been shown to be old in the art. . . ."

3. Page 6, beginning on the tenth line from the bottom it is stated as follows "The so-called limitations expressed in Claim 1 are so conventional they would have been the exceptional computer at the time of Applicants' filing date which would not include such limitations."

Finally, the Examiner states on page 7, beginning at line 8 from the bottom as follows "Applicants' contribution to the art resides in the specific instructions found within the memory locations and not the combination of three memory locations labeled first, second and third means."

Thus, the Examiner appears to be utilizing, anticipation and obviousness arguments, not issues on this appeal and without citing references, in an attempt to delineate what is new and what is old in the claim; thus having dismembered of claim (contrary to the adjudicated decisions to be discussed infra), attempts to fit the claim into a procrustean bed of 35 USC 101, instead of 35 USC 102 or 103 wherein all the facts fit.

It is submitted that by such procrustean reasoning, objects may be dissected to a point where each element of each object can be categorized for example as an arm, a leg, an eye, a tail, etc. Yet when one then tries to resynthesize the original object from an aggregate of such categorized objects as arms, legs, eyes, tails, taking any one arm or leg, etc. of various objects in accordance to category but without distinguishing the source, one doesn't obtain the original object. In a similar vein, to focus attention on only one of a claim element, as the Examiner has done and to lightly dismiss the others as inconsequential to Applicants' contribution to the prior art, is to mischaracterize the claim. The Applicants' contribution to the prior art is correctly interpreted and delineated from the prior art only when the subject matter is taken as a whole. (See infra). Only when all the elements of a claim, and how they relate, cooperate and function with each other to solve a particular problem, are considered as a whole, can a claim be properly evalu-

ated; and it makes no difference whether such elements are individually old in the art.

ARGUMENTS

Since there are no issues pertaining to anticipation or obviousness, the Appellants need not address this issue at the present time but will address the issue of whether or not it is proper to dismember claims and focus only on a small portion of the claim without treating the subject matter of the claim as a whole.

In In Re Wynne and Cousen, 118 USPQ 306, the Court stated "It is immaterial that claimed steps may be old individually where, by the claimed combination of steps, Applicant solves problem and produces desirable results not heretofore obtainable . . . claim is allowed since it cannot be said that soundness of rejection is clear beyond doubt; doubts in such cases are resolved in favor of the Applicant."

In In Re Warner and Warner, 154 USPQ 173, it is stated "Patentability of an invention is not viewed with hindsight; whether individual elements of combination are old begs the question; thus, where invention sought to be patented resides in combination of old elements, inquiry is whether bringing them together was obvious, not whether one of ordinary skill, having the invention before him would find it obvious through hindsight to construct the invention from elements of prior art." In that same case it is further stated "Language of 35 USC 102 that a person shall be entitled to a patent unless, concerning novelty and unobviousness, places burden of proof on Patent Office to produce factual bases for its rejection under Section 102 and 103; where such proof is lacking there is no necessity for resolving doubt in favor of Office's position."

In In Re Aufhauser, 158 USPQ 351, the Court said "*Concept of 'invention as a whole' comprises the total inventive act; concept includes not only the remedy claimed, but also the discovery of the source of trouble despite teachings of prior art; when considering what constitutes Applicant's 'invention as a whole' within con-*

text of 35 USC 102, Court must consider first that which Applicant discovered as cause of trouble which persisted despite the prior art, and then consider Applicant's remedy." (underlining added)

In a more recent decision published in the 11/18/76 issue of BNA's Patent, Trademark & Copyright Journal, reporting on Clopsay Corp. v. Blessings Corp., decided 10/21/76, it was stated "While the creation of a plastic film embossed with a perpendicular design may not seem like a 'substantial advance' over a previous film embossed with a diagonal design, the discovery that the new design prevents the edges of the film from curling represents a patentable advance in the eyes of the U.S. District Court for Delaware. 'It is a well established principle' says Judge Stapleton, 'that one may add to the sum of useful knowledge by uncovering the nonobvious source of a problem and in devising a solution which, though involving a combination of *old elements obvious after the source of the problem is isolated,* has never before occurred to those skilled in the art.' "

In In Re Warner and Warner, 154 USPQ 173, it is stated "Patentability of invention is not viewed with hindsight; whether individual elements of combination are old begs the question; thus, where invention sought to be patented resides in combination of old elements, inquiry is whether bringing them together was obvious. . . ."

Clearly the adjudicated decisions establishes that the subject matter of a claim must be treated as a whole and in some instances include the recognition of a problem which had eluded the prior art. Clearly the Examiner has not done this. He has dissected the claim language and eliminated the majority of elements in order to force-fit the invention into a category he believes the invention should be in and not to the category that the invention actually is in, as defined by the claims in light of the specification. It is submitted that his interpretation of the claim is erroneous. Moreover, the Appellants do not agree with him, as stated on page 4, line 19, of the Examiner's answer that "It would appear then that Applicants and the Examiner agree as to the nature of Applicants'

contribution to the art and that the portion of Claim 1 which sets forth the subject matter deemed to distinguish over that which is old in the art is the portion defining the 'data structure.'" The elements of the first claim are discussed in the Brief for the Appellant, beginning on line 21, of page 9, through line 27 of page 11. Clearly, nothing but hardware is claimed in all of Claim 1. Whether the structure claimed is old begs the question since anticipation or novelty are not issues on this appeal. Similarly each and every claim remaining in the application claims nothing but hardware. The contribution of the Applicants' invention to the prior art is clearly and succinctly set forth in the Brief for the Appellant, beginning at the bottom line of page 3 through and including line 6 of page 5 and moreover such contribution is further delineated in the claims.

The Examiner further mischaracterizes the nature of the Applicants' invention on page 7, beginning on line 20 wherein he states "The three means recited as comprising the improvement are recited as means for generating signals to cause the computer to perform certain functions. These means must be construed to read upon memory locations comprising instructions required to command and control the computer elements for performing the function recited. Applicants' contribution to the art resides in the specific instructions found within the memory locations and not the combinations of three memory locations labeled first, second and third means." It is abundantly clear that memory locations do not generate signals. What generates signals is typically the hardware depicted in Figure 13b. The micro-operation signals of Figure 13b are designated by numerals 360-365 and are mainly generated by the control unit (CSU 1301), the control store array 1333, the micro-op decoder 1359, and the logic circuitry designated α through ω. This is all hardware. A permanently microprogrammed ROM (read only memory) in the control store array 1333 is known as the firmware. It is typically comprised of large scale integrated (LSI) semiconductory circuitry. Firmware is defined in the *Computer Dictionary and Handbook*, by Charles J. Sippl and Charles P. Sippl on page 186 as "logic circuits in read-only memory. . . ."

Figures 13a and 13b are described with great particularity beginning on page 43 through 53 of the specification. Moreover, beginning on page 55, an example is taken by applying the microprogram flow chart of Figure 14a and actually microprogramming the ROM in the control store of Figure 13b to selectively generate the signals via the means claimed in the second claim for controlling certain predetermined tasks and functions. For a further discussion on this topic, see pages 10 and 11 of the Brief of Appellant. See also pages 12 and 13 of the Brief of the Appellant which gives further information as to the nature of instructions and how they are embodied in computers.

The Examiner has further mischaracterized the nature of an operating system endowing it with attributes it does not possess such as "registers." On page 5 of the Examiner's answer, lines 2-6, he states "Applicants contend that their 'firmware' invention when loaded into the registers of the operating system result in an invention having structural aspects and hardware limitations." Sippl and Sippl in their *Computer Dictionary and Handbook*, define "operating system" on page 300, as ". . . a part of a software package (program or routine) defined to simplify housekeeping as input/output procedures, sort-merge generators, data-conversion routines, or tests." Clearly an "operating system" does not have (nor can it have) registers. The correct statement is that when firmware is *incorporated* into the computer system (not operating system) it results in an improved hardware structure (Figure 13b) which cooperates and functions with other hardware structures (Figures 12, 13a) to provide new and unobvious capability to the computer system. The operating system enters into the picture only peripherally in that it can now communicate via this hardware invention with processes executing on the computer system, without programmer intervention.

Having established that the subject matter of an invention claimed must be treated as a whole, the principle enunciated by the Supreme Court in Gottschalk v. Benson 1515 USPQ 673 (1972) as interpreted by CCPA decisions is inapplicable to this case. The latest case

(of which Appellants are aware) decided on July 28, 1977 by the USCCPA, is "In the Matter of the Application of Gerald Waldbaum", re Patent Appeal 76-690. Although the Board of Appeals was affirmed since the claims were method claims in this case, the interpretation of Benson and Tabbot enunciated therein, is nonetheless significant. In the Court's opinion the preemption or nutshell rationale of Benson and Tabbot is applied to affirm the Board of Appeals. The Court defines the "nutshell rationale" as follows" . . . a patent on the claims in issue "would wholly preempt the mathematical formula and in practical effect would be a patent on the algorithm itself' (Benson supra at 72, 175 USPQ 676)." The Court further sheds light on In Re Christensen 178 USPQ 35 (CCPA 1973), wherein the Court states ". . . although the claims were limited to a specific application of technological significance, they were, nonetheless, directed to a *calculation* and would have preempted use of the algorithm in making the calculation."

The instant invention does not involve nor does it claim a method, process, or algorithm directed to a calculation. It is hardware in a computer, which permits direct communication of an operating system with processes executing on the computer system. It is disclosed as such. It is claimed as such. All that could be preempted when all the elements of a claim, with all their limitations, are taken as a whole, is the *hardware* defined by the claims that permits such communication. It does not preempt other hardware, not so defined by the claims that communicate between an operating system and process executing on the computer system. It does not preempt any algorithm for such communication because none is disclosed and none is claimed. It does not preempt any algorithm directed to a calculation. As a matter of fact, the claims when properly read considering the elements as a whole and how they function and cooperate to achieve a solution to a problem plaguing the prior art, particularly point out and distinctly claim subject matter which does not fall under the purview of 35 USC 100 or 101.

It is well established that an invention is defined by the claims interpreted in light of equivalents and the specification. It is further well established that a claim can define an invention in terms of a combination of old elements which are so related one to the other, and function one with the other to produce a new and unobvious result. It is also established that such a claim, or any claim for that matter, must be interpreted as a whole, including many times the solution to the problem which eluded the prior art. To dissect such a claim and eliminate elements from consideration on the pretext of being old in the art, and in order to force fit it in a category completely alien to its own attributes, is not supported by either statutory or case law.

It is submitted that every invention is *based* on a fundamental law, basic principle, mathematical formula, or algorithm. 35 USC 101 proscribes patenting the fundamental law, basic principle or mathematical algorithm on which the invention is based. To do otherwise would preclude meritorious future inventions based on such fundamental law, basic principle or mathematical algorithm. Benson & Tabbot admonishes us that when the language of a claim is so broad so as to preempt all uses of an algorithm, it is in effect a claim to the algorithm itself. By arbitrarily eliminating essential and limiting elements from consideration even though they be old in the art, a claim can arbitrarily be interpreted so broadly so as to read on the algorithm itself. This was not done in Benson & Tabbot and it should not be permitted here.

CONCLUSION

The sole issue in this appeal is whether or not the subject matter claimed is statutory under 35 USC 101. Clearly the subject matter of the invention is in the technological arts. It is not a method, technique or process nor claimed as a method, technique or process. It is not an algorithm nor is an algorithm claimed. It does not preempt any or all algorithms utilized to communicate between an operating system of a computer system and processes executing on the computer system. It is not an

algorithm directed to a calculation. The subject matter of the invention as disclosed and claimed is hardware. Clearly the subject matter of the invention is patentable under 35 USC 101, and it is respectfully requested that the Examiner be reversed, the claims allowed, and the application passed to issue.

ORAL HEARING REQUEST

Appellants have requested an Oral Hearing, accordingly two extra copies of this Reply Brief are provided in accordance with Rule 192.

Respectfully submitted,

/s/ Nicholas Prasinos
NICHOLAS PRASINOS
Registration No. 24,791

OPINION AND DECISION OF BOARD OF APPEALS, SEPTEMBER 20, 1978.

Before Burns and Spencer, Examiners-in-Chief, and Craig,

Acting Examiner-in-Chief.

Burns, Examiner-in-Chief.

This is an appeal from the final rejection of claims 1-6, constituting all of the claims presently in the case.

The subject matter concerns computer "firmware." [1] In their brief appellants characterize their invention in the following words:

> ". . . More specifically, the invention concerns a combination of hardware including a model-independent instruction means which is implemented in firmware, for adding capability and increased functionality to the computer system. As defined by the Applicants' claims, the present invention is directed to a combination of hardware structures controlled by a microprogrammed control system which provides for the generation of signals which cause the creation of an image of a communication center at another location, and for the generation of signals which causes the interchange of the information between the communication center and its image."

More particularly, the problem addressed and appellants' solution thereto may be discerned from the following selected portions of appellants' disclosure. From page 4, beginning at line 15:

> "High performance computing systems find it necessary to store portions of the system base in internal scratchpad registers instead of main memory,

---

[1] The term has been defined as follows:

firmware-Logic circuits in read-only memory that may be altered by the software under certain circumstances, (*Computer Dictionary and Handbook*, Sippl and Sippl, published by Howard W. Sams & Co., Inc., The Bobbs-Merrill Co. Inc., 2nd Ed., 1972.)

and hence inaccessible and invisible to software running in the system. Accordingly, a problem results when software (i.e. the programmer) wants to change positions or change information of the system base which happens to be located in these internal central processor scrathpad [sic] registers which software cannot access."

From the same page, beginning at line 24:

"The problem has been solved in one of two ways. The first method is to simply re-initialize the entire system, loading the system base with different contents as a result of a new initialization process. This technique is extremely time-consuming and quite undesirable.

"A second technique for solving the problem is to have the software employ special model-independent properties of the particular machine on which it was running, to change the system base. This results in the very undesirable property of completely model-dependent software—a feature which certainly does not serve to endear a computer manufacturer to a customer.

"What is needed therefore is a technique of changing information in the system base both in main memory and internal scratchpad registers making the system base in scratchpad memory accessible to software but not visible to software i.e. cannot be read by the software."

From page 5, beginning at line 26:

"A model-independent instruction in firmware is provided to change, upon its execution, information in the system base which is located in both main memory and internal scratchpad registers.

"The system base located at a location specified by the boundary address register BAR in main memory, is copied into a new location in main memory with predetermined portions of its information changed. This is readily accomplished by the operating system. . .

. . .

> The switch system base instruction then transfers the information which is in the system base at the new location into the system base at the location specified by BAR and also that portion of the system base located in scratchpad memory."

We reproduce claim 1 as representative, however, in so doing, we interleave indented explanation (identified parenthetically with capital letters A to D) of the respective portions of the claimed subject matter, the explanation being derived from pages 9 and 10 of appellants' brief

1. In a multiprogramming computer system having a main memory, a central processing unit (CPU) coupled to said main memory, said (CPU) controlling the state of a plurality of groups of processes being in a running, ready, wait or suspended state, said computer system also having scratchpad registers being accessible to an operating system for controlling said multiprogramming computer system, a data structure for storing coded signals for communicating between said processes and said operating system, and said scratchpad registers, said data structure comprising:

(a) first means in said data structure and communicating with said operating system for storing signals indicative of an address for a selected one of said processes;

> (A) ". . . data structure claimed is comprised of the first means which is the structure of the running process word register at BAR+56 of the system base structure of Figure 6. The system base structure is described with the particularity of 35 USC 112 beginning on page 35 through 41. Its use is further described beginning on page 41. The running process word register structure is further shown with particularity on Figure 11j and is described with particularity beginning on page 39 of the specification . . ."

(b) second means in said first means for storing coded signals indicating priority of said selected one of said processes in relation to others of said processes for obtaining control of said CPU when ready;

(B) ". . . The second means which is an element within the first part of the running process word register and is identified with reference to numeral 1129. . ."

(c) third means in said data structure and communicating with said operating system, for storing coded signals indicative of an address for a selected one of said plurality of groups of processes; and,

(C) ". . . The third means within the data structure is the J table word means and the G table word means at BAR and BAR+4 on Figures 11a and 11b and described in the specification. . ."

(d) forth means coupled to said data structure and said scratchpad registers, for generating signals causing the changing of information in said data structure and said scratchpad registers.

(D) ". . . Finally, the fourth means claimed is shown on Figures 13a and 13b is a central store as microprogrammed by the microprogram as disclosed by the flow charts of Figures 15b and 15c. Details of the control store which is microprogrammed are shown on Figure 13b (it should be understood that control stores are old in the art and persons of ordinary skill in the art can microprogram them when provided with the fiow charts of the microprogram and this results in the firmware. . ."

THE REJECTION

All of the claims have been rejected under 35 USC 101 as drawn to non-statutory subject matter. The examiner cites *Gottschalk* v. *Benson*, 409 US 63, 175 USPQ 673 (1972).

In his answer, beginning at page 5, line 6, the examiner explains his position in part as follows:

". . . Claim 1 has been couched in apparatus format and recites first means through fourth means. The first means through third means set forth means for storing signals and read upon registers or memory

locations within the operating system, already shown to be old in the art. The function of or attribute of the signals stored in the first, second, and third means for storing may be classified as indicating an address for a selected one of a plurality of processes, indicating a priority of said selected process relative to other processes, and indicating an address of a selected one of the plurality of processes, respectively. There is clearly no novelty to be found here for distinguishing over that which is deemed to be old in the art.

"The fourth means is stated as means for generating signals causing the changing of information in the data structure and scratchpad registers and is categorized by applicants' brief, page 10, as a control store as microprogrammed by the microprogram of Figures 15b and 15c. The use of control stores in computer systems has been shown to be old in the art in the listing of publications and U. S. Patents in paragraph 19 of Appendix I, amendment filed August 11, 1976. In fact, paragraph 22 of the Appendix I correctly indicates that microprogrammed computer processors including microprogrammed control units are old in the art and suggests that the systems shown in the two cited patents are capable of practicing or implementing the instant invention. . . .

. . . .

Applicants' invention resides in an instruction in firmware which changes, upon execution, information in the system base. The claim is thus reciting prior art coupled with subject matter which the U.S. Supreme Court has found to be non-statutory in *Benson*. . . ."

The appellants' argument is to the effect that only hardware, that is, structure, has been recited wherefore the *Benson* decision is inapposite. They characterize the examiner's rationale as procrustean "in forcing an arbitrary conclusion or solution to the facts," brief, bottom of page 8.

OPINION

In order to determine what appellants regard as their invention, it is necessary to consider the claimed subject matter as a whole, *In re* Noll, 545 F.2d 141, 191 USPQ 721 (CCPA, 1976), *In re* Chatfield, 545 F.2d 152, 191 USPQ 721 (CCPA, 1976). With this caveat in mind it is nevertheless appropriate to consider the features, both novel and otherwise, within the claims, as did the Court in *Parker* v. *Flook*, 437 US ——, 198 USPQ 193 (1978), an approach "not at all inconsistent with the view that a patent claim must be considered as a whole."

Analysis of Claimed Subject Matter

First, we observe that neither the term "firmware" nor an apparent equivalent appears in any of the claims, cf. 35 USC 112, last sentence. The various recited "means" for storing, communicating and generating are not to be construed as limited to firmware.

In an amendment filed August 11, 1976 (Paper No. 4), appellants amended their specification with APPENDIX I providing numbered items which refer to prior art features, to which APPENDIX we will have occasion to refer.

Our comments on the elements recited in the claims will follow their order of recitation within each claim.

Claim 1

For the main memory, appellants in APPENDIX I, (11) (a) appear to rely for disclosure upon references 2-7 found on page 2 of the specifications. References 2-6 upon inspection appear to show a mix of block diagrams directed to both hardware and software. Reference 7 shows only flow charts without a suggestion of hardware or apparatus structure. To the extent that references 2-6 disclose structure or hardware it would appear that, as such, these references demonstrate that memory *per se* is old. It does not appear that appellants have argued otherwise, rather, appellants urge that they claim ". . . new, novel and unobvious *arrangement* of these struc-

tures . . .," brief, page 9, beginning at line 4, (our emphasis).

As for the central processing unit (CPU), no specific reference has been identified in APPENDIX I by this term, nevertheless, item (22) thereof describes microprogrammed computer processors employing microprogrammed control units "with which the instant invention may be practiced."

Concerning the "data structure" of the claim, page 11, beginning at line 18, of the specification characterizes this feature of being composed of data formats, software visible registers, and instruction formats. The first and last of these three features are described in paragraphs A of specification page 11, line 21 and J of page 14, line 15. Nothing in this description of formats suggests hardware. It would seem that reliance must be placed on the software visible registers, the second feature, as described in paragraph E, page 12, beginning at line 24 of the specification. Appellants appear to rely at prior art for such features upon patents described in (12) and (13) in APPENDIX I.

Appellants have characterized the "first means" as the running process word register within the data structure. Save for the actual word stored which, it would appear, will change depending upon the nature of the data processing undertaken, this feature consists of space in memory (BAR+56 as shown in Figure 6) as part of the system base.

From the foregoing it would seem that the data structure consists of at least one software visible register, and the first means in said data structure constitutes that register (or a portion of it) with a particular word stored therein, an address. It does not appear that the "first means" constitutes novel hardware distinct from the registers designated above representing the data structure for which appellants have referenced prior art.

Similarly, for the second and third means of the claim, so much thereof as may be characterized as hardware would seem to have antecedents in the prior art registers as part of memory described above for the data structure.

The fourth means of the claim has been designated by the appellants as a central store, known to be old, brief, page 10, line 8. It is there characterized:

> ". . . as microprogrammed by the microprogram as disclosed by the flow charts of figures 15b and 15c . . ."

Appellants have buttressed this by the prior art references in APPENDIX I (19), (20) and (21).

From the above we conclude that the components recited in claim 1 appear to be old in the art constituted as registers as part of memory (the latter being hardware) save for the microprogramming provided by element (d) of the claim, the fourth means. Any "specific new, novel and unobvious arrangement of these structures," brief, page 9, beginning at line 4, would appear to be provided by the microprogramming of Figures 15b and 15c, which together with its attendant memory hardware appears to constitute firmware.

Claims 2-6

As to these claims, analysis of the nature set forth above in connection with claim 1 leads to a similar conclusion. We would only add that, with respect to claim 4, appellants have made reference to arithmetic and logic units in Appendix I(2) (a). The element characterized as "hardware gate" found in claim 5 appears also to be part of memory, for which appellants have provided prior art references, under the control of the programming shown in Figures 15b and 15c, see specification page 64, beginning at line 26 et seq., of the specification.

Authorities

Appellants in their brief rely upon *In re* Noll, *supra*, and *In re* Chatfield, 545 F.2d 152, 191 USPQ 730 (CCPA, 1976). In their reply brief, appellants comment upon *In re* Waldbaum, 559 F.2d 611, 194 USPQ 465 (CCPA, 1977) and urge the significance of the court's language in distinguishing the Waldbaum subject matter from that in *In re* Christensen, 478 F.2d 1392, 178 USPQ 35 (1973). Appellants there maintain that "the instant in-

vention does not involve nor does it claim a method, process or algorithm directed to a calculation." We disagree. It appears to us that Waldbaum, if anything, supports the examiner's position, in that the Court proscribed claims "which are directed to methods for controlling the operation of a data processor," basing its decision upon the *Benson* decision. It would seem that appellants' claims are similarly directed.

We recognize that method claims were under consideration in Waldbaum whereas here the claims are in apparatus format. In *In re* Freeman, 197 USPQ 464 (CCPA, 1978) the Court found such a circumstance to be of no moment:

> "Though a claim expressed in 'means for' (functional) terms is said to be an apparatus claim, the subject matter as a whole of that claim may be indistinguishable from that of a method claim drawn to the steps performed by the 'means.'
>
> "We agree with the solicitor's contention that if allowance of a method claim is proscribed by Benson, it would be anomalous to grant a claim to apparatus encompassing any and every 'means for' practicing that very method."

We are aware that the Waldbaum claims included characteristic language obviously related to calculating and mathematical problem solving, thus (from *Waldbaum* claim 1):

> ". . . a method . . . to *count* the number of busy lines. . . . . *comparing* means . . . to derive the *number of 1's* in said data word." (Emphasis added.)

That appellants' claim are absent similar characteristic language does not make the functions attendant the "means" of appellants' claims any less mathematical or less related to an algorithm within the meaning assigned that term by the USSC in *Benson.*

The specification states, page 12, line 2:

> "All data are in binary form, but may be interpreted as binary, decimal or alphanumeric."

Since digital computers normally operate in some numerical radix, binary, binary coded decimal, or the like, we consider the operation of appellants' claimed invention to be mathematical.

Every operation performed in appellants' invention as claimed involves the accommodation of data and instructions to the size of the registers in memory, and to the positional assignment to the registers in memory by the use of some numerical measure or quantity effected by way of electrical signals. In whatever form the instructions employed in appellants' invention are characterized, numerical or otherwise, we think it is accurate to say that the operation of appellants' structure is mathematical and that the instructions constitute a procedure which is algorithmic in character, to the same degree as that of the Waldbaum structure and that satisfactory operation of the apparatus claimed represents the successful solution of a mathematical problem falling within the definition of algorithm supplied in *Benson* and reiterated in *Flook*.

Here the method of calculation is to be used in the internal operation of the computer system and the claims recites such an end use by means of the functions recited for the various means. A claim for an improved method of calculation, even when tied to a specific end use, is unpatentable subject matter under 35 USC 101, *Flook*, *supra*.

SUMMARY

We agree with the examiner. The claims under consideration, when each is considered as a whole, recite apparatus as a matter of form only, and couple such apparatus, which appellants have indicated as old in the art, with subject matter, namely, programming, which is nonstatutory under the *Benson*, *Christensen*, *Waldbaum* and *Flook* cases, *supra*. Employing appellants' metaphor, the claims at bar, we think, extend beyond the procrustean statutory bed.

The decision of the examiner is affirmed.

AFFIRMED

/s/ [Illegible]
Examiner-in-Chief

/s/ [Illegible]
Examiner-in-Chief

/s/ [Illegible]
Examiner-in-Chief
(Acting)

BOARD OF APPEALS

NOTICE AND REASONS OF APPEAL TO UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS, NOVEMBER 17, 1978.

November 13, 1978

Commissioner of Patents & Trademarks
Washington
District of Columbia 20231

Dear Sir:

Applicants in the above-identified appeal hereby serves notice under 35 USC 141 and 142 of their appeal to the Court of Customs and Patent Appeals from the decision of the Board of Appeals in Appeal No. 329-86 rendered on September 20, 1978. The following are assigned as reasons of appeal:

The board erred in holding that:

1. It is appropriate to consider the features, both novel and otherwise within the claims, when only a non-statutory subject matter rejection under 35 USC 101 was at issue.
2. Any specific new, novel and unobvious arrangement of the combination claimed would appear to be provided by the microprogramming of figure 15b and 15c, which together with its attendant memory hardware appears to constitute firmwave.
3. Since digital computers normally operate in some numerical radix, binary, binary coded decimal, or the like, we consider the operation of appellants' claimed invention to be mathematical.
4. In whatever form the instructions employed in appellants' invention are characterized, numerical or otherwise, we think it accurate to say that the operation of appellants' structure is mathematical and that the instructions constitute a procedure which is algorithmic in character, to the same degree as that of Waldbaum structure and that satisfactory operation of the apparatus claimed represents the successful solution of a mathematical

problem falling within the definition of algorithm supplied in Benson and reiterated in Flook.

5. Here the method of calculation is to be used in the internal operation of the computer system and the claims recite such an end use by means of the functions recited for the various means.
6. A claim for an improved method of calculation, even when tied to a specific end use, is unpatentable subject matter under 35 USC 101.
7. The claims under consideration, when each is considered as a whole, recite apparatus as a matter of form only, and couple such apparatus, with subject matter, namely, programming, which is non-statutory under the Benson, Christensen, Waldbaum and Flook cases.

The board erred in not reversing the examiner's holdings that:

1. Claims 1-6 are deemed properly rejected under 35 USC 101 as drawn to non-statutory subject matter.
2. Applicants' invention resides in an instruction in firmware which changes, upon execution, information in the system base. The claim is thus reciting prior art coupled with subject matter which the U.S. Supreme Court has found to be non-statutory in Benson.
3. The claims are reciting prior art coupled with subject matter which the U.S. Supreme Court has found to be non-statutory in Benson.
4. Applicants' contribution to the art resides in the specific instruction found within the memory locations and not the *combination* of three memory locations labeled first, second, and third means.

Respectfully submitted,

/s/ Nicholas Prasinos
Nicholas Prasinos
Attorney for Applicants
Registration No. 24,791

REJECTED CLAIMS.

1. In a multiprogramming computer system having a main memory, a central processing unit (CPU) coupled to said main memory, said (CPU) controlling the state of a plurality of groups of processes being in a running, ready, wait or suspended state, said computer system also having scratchpad registers being accessible to an operating system for controlling said multiprogramming computer system, a data structure for storing coded signals for communicating between said processes and said operating system, and said scratchpad registers, said data structure comprising:

(a) first means in said data structure and communicating with said operating system for storing coded signals indicative of an address for a selected one of said processes;

(b) second means in said first means for storing coded signals indicating priority of said selected one of said processes in relation to others of said processes for obtaining control of said CPU when ready;

(c) third means in said data structure and communicating with said operating system, for storing coded signals indicative of an address for a selected one of said plurality of groups of processes, and,

(d) fourth means coupled to said data structure and said scratchpad registers, for generating signals causing the changing of information in said data structure and said scratchpad registers.

2. In combination with a communication center located in a predetermined area of a computer main memory the computer having a central processing unit (CPU) for executing plurality of processes that compete with each other for control of said CPU, operating system means coupled for providing information to said communication center, and a scratch pad memory in said CPU which stores, under control of said operating system means, portions of information located in said communication center, an improved system controlled by one of said processes for changing information in said communica-

tion area and in said scratchpad memory, said improvement comprising:

(a) first means for generating signals causing the creation of an image of said communication center at another location in said main memory;

(b) second means responsive to said operating system means for generating signals causing a change of the information in said image, resulting in a new image; and,

(c) third means, coupled for being responsive to said first and second means and to one of said processes for generating signals for causing a transfer of the new image to said communication center and to said scratchpad memory.

3. The system as recited in Claim 2 wherein said means is a switch system base instruction in firmware.

4. The system as recited in Claim 3 further including an arithmetic and logic unit having an auxiliary memory said arithmetic and logic unit coupled for communicating with said first, second and third means.

5. The system as recited in Claim 4 wherein said auxiliary memory includes a hardware gate which has a locked and unlocked state said unlocked state permitting internal interrupts or external exceptions, and said locked state not permitting internal interrupts or extral exceptions.

6. The system as recited in Claim 5 wherein said hardware gate is locked by said switch system base instruction when executing and unlocked by said switch system base instruction when it terminates execution.

SUPREME COURT OF THE UNITED STATES

No. 79-855

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS, PETITIONER

v.

JOHN J. BRADLEY and BENJAMIN S. FRANKLIN

ORDER ALLOWING CERTIORARI

Filed March 17, 1980

The petition herein for a writ of certiorari to the United States Court of Customs and Patent Appeals is granted. The case is set for oral argument in tandem with No. 79-1112, *Diamond* v. *Diehr, et al.*

☆ U. S. GOVERNMENT PRINTING OFFICE: 1980 317794 282

Supreme Court, U. S.
FILED
JAN 22 1980
MICHAEL RODAK, JR., CLERK

IN THE

Supreme Court of the United States

OCTOBER TERM, 1979

No. 79-855

SIDNEY A. DIAMOND, COMMISSIONER
OF PATENTS AND TRADEMARKS

Petitioner,

v.

JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN

Respondents.

BRIEF IN OPPOSITION OF A WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

Of Counsel:

FAITH F. DRISCOLL
HENRY L. HANSON
RONALD T. REILING

NICHOLAS PRASINOS
Honeywell Information Systems, Inc.
Office of the General Counsel
200 Smith Street
Waltham, Massachusetts 02154
(617) 895-6129

Counsel for Respondents

Washington, D.C. • CLB PUBLISHERS' • LAW PRINTING CO. • (202) 393-0625

TABLE OF CONTENTS

Page

INTRODUCTION 1

COUNTERSTATEMENT OF QUESTION PRESENTED 5

CONSTITUTIONAL AND STATUTORY PROVISIONS INVOLVED 5

STATEMENT OF THE CASE 6

ARGUMENT 10

SUMMARY 19

CONCLUSION 21

APPENDIX A 1a

APPENDIX B 1b

APPENDIX C 1c

TABLE OF AUTHORITIES

Cases:

Bradley and Franklin, In re, 600 F.2d 807 (1979) 9

Chatfield, In re, 545 F.2d 152 (1976) 4

de Castelet, In re, 562 F.2d 1236 (1977) 18

Diehr, In re, 602 F.2d 982 (1979) 18

Eibel Process Co. v. Minnesota & Ontario Paper Co., 261 U.S. 45 (1923) 13,16

Funk Brothers Seed Co. v. Kalo Innoculant Co., 333 U.S. 127 (1948) 11

Gelnovatch, In re, 595 F.2d 32 (1979) 18

Gottschalk v. Benson, In re, 409 U.S. 63 (1972) ... 2,3,9,11,19

Graham v. Deere, 383 U.S. 1 (1966) 13,16

Johnson, Parrack and Lunsford, In re, 589 F.2d 1070 (1978) 4,18

Mackay Radio & Telegraph Co. v. Radio Corporation of America, 306 U.S. 86 (1939) 11,16

Maucrops, In re, No. 79-554 (CCPA, November 1, 1979) 17

Parker v. Flook, 437 U.S. 584, 98 S.Ct. 2522 (1978) ..... *passim*

Richman, In re, 563 F.2d 1026 (1977) 18

Safety Car Heating & Lighting Co. Inc, v. General Electric Co. et al 155F.2d 937 (1946) 15

Sarkar, In re, 588 F.2d 1330 (1978) 18

U.S. v. Adams, 383 U.S. 39 (1966) 14,16

Waldbaum, In re, 559 F.2d 611 (1977) 18

Statutes:

35 USC 101 ..... *passim*

35 USC 112 12,17

Miscellaneous:

Rules of Practice in Patent Cases 12

Computer Dictionary and Handbook (2d ed. 1972), C. J. Sippl and C. P. Sippl 8,19

IN THE

Supreme Court of the United States

OCTOBER TERM, 1979

No. 79-855

SIDNEY A. DIAMOND, COMMISSIONER
OF PATENTS AND TRADEMARKS

Petitioner,

v.

JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN

Respondents.

BRIEF IN OPPOSITION OF A WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

INTRODUCTION

There is no basis for issuance of a Writ of Certiorari in this case. None of the criteria which normally are considered in the Court's discretion are present here.[1] In particular, there is no conflict with any applicable decision of this Court.

[1] See Supreme Court Rule 19.

The machine invention which the Respondents seek to patent has been mistakenly represented by the Petitioner to be a computer program when in fact it is clearly not a computer program.

The Respondents strongly submit that whether computer programs per se are patentable is not an issue in this Petition. It should be noted that the real party in interest, the Assignee of the Respondents' Application, has been on record before this Court as opposing the patentability of computer programs.[2] The Respondents' invention is not a computer program; if it were such a program, no patent application would have been filed. Furthermore, Respondents and others are of the view that computer programs are more properly the subject matter of copyright.[3]

The invention claimed by Respondents is a new computer machine (R6-111 and R207-208).[4] The United States Court of Customs and Patent Appeals unanimously found that the invention was a machine (PW-15a).[5]

---

[2] See Brief of Honeywell Inc. Amicus Curiae in Gottschalk v. Benson, 409 US 63, (1972).

[3] See Final Report of the National Commission on New Technological Uses of Copyright Works. The Commission on page 1 has recommended to President Carter and to the U.S. Congress that:

> "The new copyright law should be amended: (1) to make it explicit that computer programs, to the extent that they embody an author's original creation are proper subject matter of copyright. . . ."

[4] The letter R followed by a numeral denotes the page number of the Transcript of the Record before the U.S. Court of Customs and Patent Appeals to which reference is made.

[5] PW indicates the page number of the Petition for Writ of Certiorari that the Petitioner has filed before this Court.

Even the Petitioner through the Associate Solicitor of the United States Patent and Trademark Office has admitted that Respondents' invention is a machine.[6] With the admission that the Respondents' invention is a machine, the issue of statutory subject matter is resolved because all the rejections and objections of the Patent and / Trademark Office have been overcome.[7] Under 35 USC 101 new machines are explicitly stated to be statutory subject matter. Hence, there is no real issue for this Court to decide.

It is submitted that by filing the Petition for Writ of Certiorari, the Commissioner of Patents and Trademarks is misleading this Court by "labeling" the invention a computer program.[8] The tactic of labeling inventions as

---

[6] See Appendix A.

[7] The question of "newness" had already been addressed in prosecution and resolved in Respondent's favor. (See PW-34a). There the Examiner cited the Hauck patent as prior art which was successfully rebutted by the Respondents and the rejection withdrawn by the Examiner with only a 35 USC 101 rejection remaining. (PW-37a and 41a). On the record before this Court, the Respondents' invention is acknowledged to be a new machine as claimed.

[8] Even if the invention were a program, which it is not, this Court has consistently refused to invade the legislative prerogative of Congress with regard to such matters. In Gottschalk v. Benson, 409 US 63, (1972) this Court stated:

> "If these programs are to be patentable, considerable problems are raised which only committees of Congress can manage, for broad powers of investigation are needed, including hearings which canvass the wide variety of views which those operating in this field entertain. The technological problems tendered in the many briefs before us indicate to us that considered action by the Congress is needed."

(continued)

computer programs and then finding such inventions unpatentable has long been followed by the Patent and Trademark Office. The United States Court of Customs and Patent Appeals has repeatedly admonished the Patent and Trademark Office about its use of this tactic.[9]

(footnote continued from preceding page)

In Parker v. Flook, 437 US 198 (1978) this Court again reiterated its position as follows:

> "Difficult questions of policy concerning the kinds of programs that may be appropriate for patent protection and the form and duration of such protection can be answered by Congress on the basis of current empirical data not easily available to this tribunal."

[9]The Court of Customs and Patent Appeals in In re Chatfield, 545 F.2d 152, 191 USPQ 730 (1976) stated:

> "The unique technological aspects of digital computers and the changes wrought in their operation by associated software programs, coupled with the limited number of precedental court decisions, illustrate the undesirability of deciding the issue of patentability by mere use of the label 'computer program.'"

In re Johnson, Parrack and Lunsford 589 F.2d 1070, 200 USPQ 199 (1978) the Court of Customs and Patent Appeals said:

> "The Solicitor argues in his supplemental brief that the claims at issue are not statutory subject matter because they are computer programs and computer programs are not patentable under Benson and Flook. This broad statement is not at all germane to the considerations before this Court and is an erroneous statement of the law. We stated in In re Chatfield, supra at 155, 191 USPQ 733 that 'the mere labeling of an invention as a computer program does not aid in decision making and the Supreme Court declined to decide either Benson or Flook on such broad nonsubstantive ground. . . .'"

COUNTERSTATEMENT OF QUESTION PRESENTED

The "Question Presented" by Petitioner reflects a clear misunderstanding of the invention and a misconstruction of the record.[10] It is based upon a false premise. The essential issue in the present case on which Petitioner seeks this Court's intervention may be accurately and properly stated as follows:

> "Is a machine invention which is in the technological arts, which is constructed and claimed as a combination of elements, patentable subject matter under 35 USC 101?"

CONSTITUTIONAL AND STATUTORY PROVISIONS INVOLVED

Article I, Section 8, Clause 8, of the Constitution of the United States provides:

> The Congress shall have Power * * *
>
> * * *
>
> To promote the Progress of Science and useful Arts, by securing for limited Times to Authors and Inventors the exclusive Right to their respective Writings and Discoveries.* * *

Petitioner has correctly stated the applicable statutory provision, 35 USC 101.

---

[10]The Examiner, the Board of Appeals and the Solicitor of the U.S. Patent and Trademark Office have repeatedly misinterpreted and misconstrued the Respondents' invention. See Appendix B for illustrative errors.

STATEMENT OF THE CASE

The United States Court of Customs and Patent Appeals correctly describes the inventive subject matter and accurately relates the reasoning underlying the application of 35 USC 101 to this inventive subject matter.[11] Respondents need not, therefore, repeat here these details, except in the most fundamental way so that a clear and undistorted picture of the nature of the invention and its judicial history is presented.

This invention is one of a series of inventions which collectively define an entirely new computer machine which is now being marketed worldwide as the Honeywell Series 60, Level 64 computer product line. The invention, as disclosed and claimed in the Respondents' application, is directed to a unique design of a portion of this Level 64 computer system. This portion which the Respondents have invented is a new and useful computer machine.[12] The new computer machine utilizes as part of this unique design, an implementation of the invention which includes firmware as one element of the new combination. This invention can also be implemented totally in a hardwired embodiment—that is without any firmware element. It is merely a design implementation choice. This type of invention if imple-

---

[11] See PW 1a-15a.

[12] A complete discription of the invention from the Record below (R3-R111) is contained in Appendix C (pps. 1c-111c) The description includes twenty-two (22) sheets of drawings, eighty-six (86) pages of specification and six (6) claims. The Petitioner in his Petition and Brief has offered, out of context, a portion of the description of the invention compromising only one (1) sheet of drawing and five (5) pages of description (PW 42a-46a) purporting to describe this complex invention.

mented totally in hardwire would clearly be statutory subject matter under 35 USC 101.[13] The invention cannot be implemented as a computer program to be executed by any general purpose computer. It is not a computer program invention.

Perhaps the nature of the invention and how it is completely different from a computer program may best be illustrated by a simple analogy. A consideration of a player piano and its operation is particularly useful in pinpointing and understanding the nature of the invention.

Player pianos and their method of operation, using rolls into which the notes of a musical composition have been punched, are well known. A player piano, once designed and operative, will respond to a musical composition encoded in a roll and fed into it by automatically playing the composition in the manner intended by the composer. The piano, once the implementation has been fixed, will always be the same combination of elements and operate in the same manner. However, the internal mechanism of a player piano may obviously be redesigned and implemented as a new combination which operates in a new and different manner. The redesigned piano is a new player piano (a new combination). Externally the new combination still performs the musical compositions recorded

[13]The following U.S. Patents are examples of hardwired implementations of similar types of inventions classified in the same technical class 340 and 172.5 as the Respondents' inventions: 3,454,933; 3,454,931; 3,425,039; 3,323,110; 3,522,589; 3,344,404; and 3,564,504. Additionally, the following U.S. Patents involve similar types of inventions implemented in firmware: 3,400,371; and 4,028,670.

in any piano roll inserted therein.

By analogy therefore the invention the Respondents are claiming is a new piano (new computer machine). They are not claiming a new piano roll (computer program). The implementation of the new internal hardware redesign is a new combination, an apparatus which is a new machine which clearly passes through the gates of 35 USC 101.

There are two separate individuals involved in the conception of the two separate products involved in the foregoing analogy. First, the composer (programmer) and second, the piano (machine) designer. While both may be very creative in their respective fields, their roles are different and distinct and the products they have created must be considered separately in terms of the application of the patent, copyright or other laws of the United States thereto.

The Respondents' computer invention is the work product of computer machine designers, (equivalent to the piano designer in the analogy). They have designed a new computer machine which operates more efficiently.

The inventors in conceiving this new machine have drawn from their creative ability and computer expertise. In implementing their invention, the inventors have utilized the latest hardware technology available which includes firmware.[14] Once the invention is incorporated into the computer machine, any compatible program

[14]The Computer Dictionary and Handbook, by Charles J. Sippl and Charles P. Sippl, published by Howard W. Sams & Son, Inc., N.Y., defines "firmware" on page 186 as follows: "Firmware—logic circuits in read-only memory. . . ."

(old or new) may be used in the new, improved computer machine. A programmer using the computer machine or writing programs therefor is normally not aware and does not have to know, how the computer machine is constructed or operates internally.

Respondents are applying for a patent covering the new combination, the new machine (new piano). They are not claiming, contrary to what the Petitioner is alleging, a computer program (piano roll). Furthermore, they are not contending that the running of a program (old or new) on the new, improved computer machine results in still another new machine; this is not under consideration here.

What the inventors have conceived is a new combination of hardware elements, which includes a firmware element. Accordingly, when any program such as a program for translating "texts from Russian to English or for determining a hitherto unknown and faster route between Washington and New York"[15] is run on this new, improved computer machine, it can be run more efficiently.

The Respondents' apparatus combination Claims 1-6 were finally rejected by the Examiner under 35 USC 101 as unpatentable and the rejection was "grounded on an erroneous interpretation of the Supreme Court's decision and opinion in *Benson*, namely, that all computer programs or program related inventions are nonstatutory under Section 101."[16] The Board of

[15]PW-12. (See also Appendix B, page 36).

[16]In re Bradley and Franklin, 600 F.2d 807, 202 USPQ 480 at 484 (1979).

Appeals affirmed the Examiner (R203) further considering the invention in view of *Parker v. Flook* (R197), 437 US 188, (1978).

The United States Court of Customs and Patent Appeals, on July 5, 1979, unanimously reversed both the Examiner and the Board of Appeals stating:

> ". . . We disapprove the Board's distillation of Appellants' claim down to the information contained in the firmware in order to hold it nonstatutory."
>
> "We find that the invention is a combination of tangible hardware elements—a machine. . . ."
>
> ". . . we do not find the invention to be nonstatutory under the authority of *Benson* or *Flook*, or under the authority of our own cases. . . ."

ARGUMENT

There is no basis for granting the petition. The decision of the court below[17] does not conflict with any decisions of this Court; it is fully in conformance with the applicable Constitutional and Statutory provisions; and, it in no way extends the scope of the patent laws.

The Petitioner relies heavily upon this Court's decision in *Parker v. Flook*, 437 US 584 (1978). It is submitted that the *Flook* decision is not even relevant to the invention under consideration here for the following reasons. First of all the Petitioner erroneously characterizes the Respondents' invention as an algorithm.[18] This Court in *Flook* and *Benson* defined an algorithm as follows:

[17]PW 1a-15a.

[18]PW 10.

> "We use the word 'algorithm' in this case as we did in *Gottschalk v. Benson*, 409 U.S. 63, 65, 175 USPQ 673, 674 to mean '(a) procedure for solving a given type of mathematical problem. . . .' "[19]

No algorithm is disclosed in the Respondents' specification, no algorithm is claimed and no mathematical problem is to be solved.

In addition, this Court in its 6-3 decision explicitly held in *Flook*:

> "Very simply, our holding today is that a claim for an improved method of calculation, even when tied to a specific end use, is unpatentable subject matter under Section 101."[20]

The Respondents' invention is a machine and no method of calculation is claimed. The Petitioner has obviously misread the *Flook* decision and its applicability to the invention involved here.

Even if the *Flook* decision were applicable, the Petitioner has chosen to interpret this Court's opinion in a very questionable manner. The Court stated:

> "Mackay Radio and Funk Brothers point to the proper analysis for this case: The process itself, not merely the mathematical algorithm, must be new and useful. Indeed, the novelty of the mathematical algorithm is not a determining factor at all."[21]

However, the Petitioner confusingly alleges a two-step analysis is required under 35 USC 101 for a "patent claim involving an algorithm." The Respondents do not claim an algorithm. Therefore, the Petitioner's alleged

---

[19] 98 S. Ct. at 2523 (footnotes).

[20] *Parker v. Flook*, 98 S. Ct. at 2528 (footnotes).

[21] Ibid (2526).

two-step analysis is not even applicable to the Respondent's machine invention.

Going even further if the Petitioner's alleged two-step analysis were to be applied to Respondents' invention, the result of such analysis is that the invention is nonetheless statutory subject matter uder 35 USC 101. The first step of Petitioner's alleged test is that one must determine whether the claim contains a principle, formula, idea or concept which is unpatentable which then must be separated from the rest of the claim. There is no such principle, formula, idea or concept in Respondents' claim. Thus it passes the first step. The second step of Retitioner's alleged test is to determine if what remains of the claim is old in the art. The Respondents have continuously maintained that what they have invented and what is being claimed is a new combination of elements, and this has never been successfully rebutted. The Petitioner is now alleging that everything in the claim is old without any support in the Record for the allegation.[22] Thus even by the

[22]The Commissioner of the Patent and Trademark Office has for the first time in the prosecution of this invention, improperly and belatedly referred to, in the Petition, the fact of common subject matter in Respondents' application and the application of Carre. (PW 8-9 footnotes). The issue is not before this Court. However, Respondents note that it is accepted practice for inventors working on a common machine to disclose their separate and distinct inventions (which are both different parts of the overall machine), by describing the total machine. Accordingly, the two patent applications include common subject matter. In view of the Patent and Trademark Office Rule 71 and 35 USC 112 of the Patent Laws, the two inventions, probably could not properly be disclosed without including common subject matter. The reference to common subject matter in the Petition raises a question of novelty which is not before this Court. The issue was raised by the Examiner during prosecution and decided in Respondents' favor. (See p. 3 supra).

application of the Petitioner's own alleged two-step analysis, the Respondents' invention is patentable subject matter under 35 USC 101.

This Court in *Flook* expressly reiterated the long standing and legally sound principle that an invention must be considered as a whole[23] and then proceeded to state:

> "Respondents' process is unpatentable under Paragraph 101 not because it contains a mathematical algorithm as one component, but because once the algorithm is assumed to be within the prior art, the application considered as a whole contains no patentable invention."[24]

In *Flook* the Respondents did not deny that the invention was a process, that the process was old, and that the novelty lay only in a claimed mathematical algorithm. It was thus established by the Court that the sequence of steps in that particular process under

---

[23]Parker v. Flook 96 S. Ct. at p. 2526.

[24]This requirement was also stated by this Court in Graham v. Deere, 383 US 1 (1966), when it noted that the invention as a whole must be viewed in determining patentability, and that the whole encompasses both conception and implementation.

This reasoning is also consistent with this Court's opinion in Eibel Process Co. v. Minnesota & Ontario Paper Co., 261 US at 45 (1923). There the patentee discovered that wrinkling and malformation of paper in a paper making machine occurred because the stock did not flow fast enough at a critical point in the process. Eibel simply increased the pitch of the path of the stock to take advantage of the force of gravity to increase the flow rate. This Court held the patent valid deciding that the patentee had made a patentable contribution. It rejected arguments that the patentee was claiming a force of gravity, and considered the invention as a whole in determining patentability.

consideration was also old. It was obvious that only one step of the process was alleged to be novel. When in accordance with this Court's reasoning the mathematical algorithm (the one step alleged to be novel) was also assumed to be within the prior art (i.e., old), then that specific process as a whole was also old. Accordingly, there was no patentable subject matter contained by the claimed process.

The Respondents' invention is vastly different than the invention considered by the Court in *Flook*.

To illustrate the error of Petitioner's reasoning by analogy, assume that one is considering the following old elements: (1) wheels, (2) shafts, (3) rope and (4) a frame.

A first combination which can be claimed is a cart that is constructed with the wheels, shaft and frame to be pulled along by the rope.

A second claimable combination is a system of pulleys constructed with the wheels, shaft, frame and rope to lift large weights.

A third combination is a steering mechanism constructed with the wheels, shaft, frame and rope to steer an airplane or a boat.

All of these combinations have the same well known elements. What gives uniqueness to each combination is the arrangement of these known elements with respect to each other and how they cooperate with each other, i.e., how they are brought together.[25] Not only has the

---

[25]This Court in U.S. v. Adams, 383 US 39 (1966) said:

"It begs the question . . . to state merely that magnesium and cuprous chloride were individual known battery elements. If such combination is novel, the issue is whether bringing them together as taught by Adams was obvious in the light of prior art."

Petitioner refused to consider the arrangement of elements in Respondents' invention and how they cooperate with each other as directed by this Court, he has not identified any prior art which would render the Respondents' specific combination invention old. Judge Learned Hand in *Safety Car Heating & Lighting Co., Inc. v. General Electric Co., et al.*, 155 F.2d 937 (1946), eloquently stated the proper analysis as follows:

> "Substantially all patentable inventions are for combinations of old elements and what counts in determining whether a patentable invention has been made is the selection, out of all their possible permutations, of that new combination which will be serviceable."

If the claimed arrangement of elements and how they cooperate with each other is ignored by the Petitioner and no prior art is cited, it is clear that he could, with the same defective logic, hold as nonstatutory any combination invention ever invented and any invention still to be invented in the future.

This case can be distinguished over the *Flook* decision in still another way. In *Flook*, the Respondent was attempting to enlarge the legislative grant by claiming a process admittedly old by utilizing an unpatentable principle. He was attempting to ". . . transform an unpatentable principle into a patentable process. . . ."[26] In this case the reverse is true. The Petitioner is attempting to diminish the legislative grant by refusing to consider the Respondents' novel combination invention as statutory subject matter. The Petitioner refuses to properly examine the Respondents'

---

[26]*Parker v. Flook*, 98 S. Ct. 2526.

invention in his attempt to eliminate patent protection for computer inventions which he erroneously labels as a program or an algorithm. The Petitioner also relies on the old crutch of administrative convenience to justify his position of diminishing the scope of patent protection available under Title 35 of the United States Code. The reasoning of the Patent and Trademark Office is contrary to the legislative intent of Congress[27] and the decisions of this Court.[28]

It is ironic that at a time when the President of the United States has announced that he plans to "develop and maintain a climate conducive to industrial innovation" and "strengthen the patent system," the Petitioner is attempting to weaken it by alleging *inter alia,* that it would be more administratively efficient to deny a patent to the Respondents' complex invention without being required to search the prior art.[29] There is

---

[27]To promote the progress of Science and the Useful Arts . . .

[28]See Graham v. John Deere Co., 383 U.S. 1 (1966); McKay Radio & Telegraph Co. v. Radio Corporation of America, 306 U.S. 86 (1939); Eibel Process Co. v. Paper Co., 261 U.S. 45 (1923); and United States v. Adams, 383 U.S. 39, 50 (1966).

[29]The following are excerpts form the text of President Carter's message to Congress October 31, 1979 outlining his plan to revitalize America's industrial base.

> "TO THE CONGRESS OF THE UNITED STATES: Industrial innovation–the development and commerialization of new products and processes–is an essential element of a strong and growing American economy. . . . Further, industrial innovation is necessary if we are to solve some of the Nation's most pressing problems."
>
> "Patents can provide a vital incentive for innovation. . . . At my direction, the Patent and Trademark Office will undertake a major effort to upgrade and modernize its

(continued)

sufficient statutory and judicial law at present to correctly decide the patentability of any computer invention and the Patent and Trademark Office should be given a signal that it must follow this established law. The Court can do this by refusing to grant the Petition for Certiorari.

The decision of the United States Court of Customs and Patent Appeals below is another example of that court's clear understanding of computer inventions, the proper application of the statute 35 USC 101 to such inventions and the decisions of this Court relevant thereto. It is submitted that the United States Court of Customs and Patent Appeals has correctly applied the applicable decisions of this Court. When nonstatutory subject matter is involved, which the Appellants allege is statutory, the United States Court of Customs and Patent Appeals properly affirms the Patent and Trademark Office.[30] When on the other hand statutory

---

(footnote continued from preceding page)

process in order to restore the incentive to patents–and ultimately develop–inventions."

"We must also develop and maintain a climate conducive to industrial innovation. The Federal Government must take the lead in creating that climate."

From Congressional Quarterly Inc., November 10, 1979 (pages 2557-2558).

[30]In the Matter of the Application of Jean E. Maucorps (CCPA No. 79-554), decided 11/1/79 the court in affirming the Board of Appeals properly recognized that even though nonstatutory subject matter was "expressed" and claimed as an apparatus the invention was not patentable under 35 USC 101. The Court clearly seeing through the subterfuge said:

"That 35 USC 112 authorizes the claiming of 'mean for' performing a function cannot rescue Appellants' claim from

(continued)

subject matter is involved, which the Patent and Trademark Office alleges is nonstatutory, the Court of Customs and Patent appeals correctly applies 35 USC 101.[31]

In view of the fact that the *Flook* decision is not applicable to this case because, inter alia, no mathematical algorithm, principle, formula or idea is involved, there is no reason to even consider In re Diehr, 602 F.2d 982, (CCPA 1979) in connection with this case.[32] Whatever decision is made by this Court in its judgment regarding the Petition to be filed by the Commissioner of the Patent and Trademark Office in the Diehr case, it should be based solely on the merits of Diehr. The inventions in the two cases are so completely different that no benefit would inure to the Court in attempting to consolidate. On the contrary, in view of the complex nature of this invention, confusion would result from attempting to combine it with any other case.

(footnote continued from preceding page)

> the requirement of Paragraph 202, because Paragraph 221 does not authorize the claiming of apparatus entirely in terms of 'means for' performing a nonstatutory method."

Other cases where the United States Court of Customs and Patent Appeals affirmed the United States Patent and Trademark Office are as follows: In re Waldbaum, 559 F.2d 611, 194 USPQ 465 (1977); In re Richman, 563 F.2d 1026, 195 USPQ 340 (1977); In re de Castelet, 562 F.2d 1236, 195 USPQ 439 (1977); and In re Sarkar, 588 F.2d 1330, 200 USPQ 133 (1978).

[31]In re Johnson, Parrack and Lunsford, 589 F.2d 1070, 200 USPQ 199 (1978). In re Gelnovatch, 595 F.2d 31, 201 USPQ 136.

[32]See PW-13.

SUMMARY

The *Flook* and *Benson* decisions are not applicable to this case.

The Respondents' invention comprises a computer hardware machine implemented as a novel/hardware including a firmware element.[33] There is no question that the invention is a machine; even the Petitioner has admitted it is a machine. The applicable statute, 35 USC 101, explicitly states that new machines are statutory subject matter.

To present even a colorable case to this Court, the Petitioner has "labeled" the invention as a program. He has continually made this and other factual statements, which are totally speculative and which are not supported by the record in this case. Many of these statements are clearly erroneous. (See Appendix B). These speculative statements cannot be substantiated before this Court by the Petitioner without previous findings of fact. The Petitioner has also tried to distill the invention to one element—the firmware—contrary to this Court's decisions. The Petitioner also wrongly implies the firmware element would operate like a program in any computer machine.[34] The firmware

---

[33]Firmware is defined as "logic circuits in read only memory. . . ." See page 186 of Computer Dictionary and Handbook, by C. J. Sippl and C. P. Sippl. Howard W. Sams & Co. Inc., Indianapolis, Ind.

[34]The invention is marketed as an integral part of a new computer machine. On the other hand, computer programs are generally embodied on punched cards, magnetic tapes, or the like, and are licensed or sold as individual products, separate and apart from the computer machine.

element cannot run in any computer machine. It can operate only in combination with the other claimed elements.

A patent granted on this invention as claimed would not protect the firmware element per se. Any person could use the firmware element alone or as part of an unclaimed combination with other elements without infringing the patent.

The Petition to this Honorable Court for a Writ of Certiorari to the United States Court of Customs and Patent Appeals is based on a false premise—that this machine is a program. The Petitioner is obviously confusing a program with a machine that is programmable. The Petitioner's arguments are therefore totally without merit and based on a wrong assumption which inevitably educe wrong conclusions. A computer machine by any other name is still a computer machine.

CONCLUSION

There is no clear issue of law presented to this Court by the Petitioner. The Petitioner is asking this Court to legislate on a fact issue for which no factual record has been developed, and is not asking this Court to adjudicate an issue of law. For this and the reasons discussed above the Petition for a Writ for Certiorari should be denied.

Respectively submitted,

NICHOLAS PRASINOS
Attorney for Respondents
Honeywell Information Systems Inc.
Office of the General Counsel
200 Smith Street
Waltham, Massachusetts 02154
(617) 895-6129

Of Counsel:

F. F. DRISCOLL
H. L. HANSON
R. T. REILING

APPENDIX A

In re Application:	)	
	)	
John J. Bradley	)	Patent Appeal
and Benjamin S. Franklin	)	No. 79-533
	)	
Serial No. 570,331	)	

In the matter of the above referenced case which was heard before the United States Court of Custom and Patent Appeals, I was instructed to obtain the tape of the oral arguments from the court, listen to and transcribe the particular portion by the Solicitor.

I hereby certify that the following is a transcription of the portion requested:

> "Do you agree with Appellant that he is claiming a machine where it starts out, 'In multiprogram computer system . . .'?"

Answer:

> "I suppose it is a machine, it is an apparatus of some sort, yes Your Honor."

Date: December 7, 1979

Signed Carolyn L. Leonard
2043 Crystal Plaza Drive
Arlington, Virginia 22202

APPENDIX B

The following erroneous statements by the Petitioner are examples of the arbitrary and speculative pronouncements for which no factual record has been developed in this case, and illustrate the confusion and false conclusions of the Petitioner regarding this invention:

Erroneous statement of Petitioner:

> "Although the original PROMs could be programmed only once, many large computers now have a "read-write" PROM, which permits the user to reprogram the control memory any number of times electrically." (PW 4 footnotes).

Correct statement of fact:

There is no such thing as a read-write PROM. A PROM by its definition is a programmable read-only memory.

Erroneous statement of Petitioner:

> "Bradley's firmware is stored in a control memory of the PROM type which can be microprogrammed by the user." (PW 5 footnotes).

Correct statement of fact:

The term "user" as used in the Record means the computer manufacturer. The type of PROM used by Respondents require physical alterations by the manufacturer of the computer to create the desired circuits. Respondents' control memory cannot be reprogrammed by anyone.

Erroneous statement of Petitioner:

> ". . . the microprogram steps are disclosed in the flow charts of Figures 15b and 15c." (PW 5 footnotes).

Correct statement of fact:

Machine designers prepare a flow chart of their hardware design before building the machine. Figures 15b and 15c are flow diagrams of machine operations of one element of the Respondents' invention. Figures 15b and 15c do *not* illustrate microprogram ***steps***.

Erroneous statement of Petitioner:

> "Thus, he [the Examiner] concluded that the invention 'resides in an instruction in firmware which changes upon execution, the information in the system base.' " (PW 5 footnotes).

Correct statement of fact:

The claimed invention is a unique combination of hardware elements which cooperate and function with each other to achieve the objects of the invention on pages 5c,6c Appendix C.

Erroneous statement of Petitioner:

> "Any 'specific new, novel and unobvious arrangement of these structures . . . would appear to be provided by the microprogramming. . . ."

Correct statement of fact:

Microprogramming cannot arrange hardware structures as claimed--only the inventors can do that. The use of microprogramming is not an issue in this case.

Petitioner erroneously implies that:

> The invention is no different than a program for determining a "faster route between Washington and New York." (PW 12).

Correct statement of fact:

The Respondents' invention is a new machine not a program. A complete reading of the specification makes this clear. (Appendix C, pps. 1c-109c). The cities of Washington and New York are known and all paths between them whether or not already discovered are in existence. Merely discovering the shortest path whether with the aid of a computer program or otherwise does not breathe "newness" to the discovery of an existing path. Such a hypothetical discovery would not be statutory subject matter under 35 USC 101 in view of this Court's reasoning in *Flook*. The Respondents' invention, however, is a new combination of elements *which combination was not known prior to its creation by the Respondents.*

APPENDIX C

APPLICATION OF JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN, FILED APRIL 21, 1975, SERIAL NUMBER 570,331, FOR SWITCH SYSTEM BASE MECHANISM.

ABSTRACT OF THE DISCLOSURE

A model-independent instruction in firmware is provided to change, upon execution, information in the system base which is located in both main memory and internal scratch pad registers.

RELATED APPLICATIONS

The following applications are incorporated by reference to the instant application.

1. "Buffer Store" invented by J.L. Curley, T.J. Donahue, W.A. Martland, and B.S. Franklin, filed on October 5, 1972 having Serial Number 295,301 and assigned to the same assignee named herein.
2. "Variable Masking for Segmented Memory" invented by Wallace A. Martland and John L. Curley, filed on October 5, 1972 having Serial Number 295,303 and assigned to the same assignee named herein.
3. "Override Hardware for Main Store Sequencer" invented by Thomas J. Donahue, filed on October 5, 1972 having Serial Number 295,418 and assigned to the same assignee named herein.
4. "Main Memory Sequencer" invented by T.J. Donahue, J.L. Curley, B.S. Franklin, W.A. Martland, and L.V. Cornaro, filed on October 5, 1972 having Serial Number 295,331 and assigned to the same assignee named herein.
5. "Main Memory Reconfiguration" invented by J.L. Curley, B.S. Franklin, W.A. Martland, T.J. Donahue and L.V. Cornaro filed on October 5, 1972 having Serial Number 295,417 and assigned to the same assignee named herein.
6. "Ring Checking Hardware" invented by _____ filed on the same date as the instant invention, having Serial Number _____ and assigned to the same assignee named herein.
7. "Segmented Address Development" invented by _____, filed on _____ and having Serial Number _____ and assigned to the same assignee named herein.
8. "Protection of Information in a Multiprogram Multiprocessor Computer System" invented by _____, filed

on _____ and having Serial No. _____ and assigned to the same assignee named herein.

9. "Ring Crossing Mechanism" invented by _____, filed on _____, and having Serial Number _____ and assigned to the same assignee named herein.

10. "Procedure Calls and Stack Operations", invented by _____, filed on _____, and having Serial Number _____ and assigned to the same assignee named herein.

11. "Process Synchronization Utilizing Semaphores" invented by _____, filed on _____, and having Serial Number _____.

12. "Arrangement for Automatically Managing, Scheduling and Allocating System Resources" invented by _____, filed on _____ and having Serial Number _____ and assigned to the same assignee named herein.

13. "Data Processing Memory Management System" invented by _____, filed on _____ and having Serial Number _____ and assigned to the same assignee named herein.

14. "Arrangement for Initializing a Computer System Controlled by Reloadable Site-Dependent Information Structures", invented by _____, filed on _____ and having Serial Number _____ and assigned to the same assignee named herein.

15. "System Base", invented by _____, filed on _____, and having Serial Number _____ and assigned to the same assignee named herein.

16. "Absolutization During Initialization", invented by _____, filed on _____, and having Serial Number _____ and assigned to the same assignee named herein.

BACKGROUND

Field of the Invention:

This invention relates generally to computer systems and more particularly to a system and method for communicating between hardware and software in a multiprogramming/ multiprocessing environment.

Multiprogramming/multiprocessing computer systems require the use of a communications area in main storage for communications between hardware and software. (See the application entitled "System Base" invented by _____ filed on _____ having Serial Number _____ and assigned to the instant assignee, and which application has been incorporated by reference herein). The system base is generally located at a fixed location in main memory, although this location may be changed or relocated through the use of a device such as a boundary address register (BAR). (See invented by _____, filed on _____, having Serial Number _____ and assigned to the instant assignee, and further incorporated by reference in the instant application).

High performance computing systems find it necessary to store portions of the system base in internal scratchpad registers instead of main memory, and hence inaccessible and invisible to software running in the system. Accordingly, a problem results when software (i.e. the programmer) wants to change positions or change information of the system base which happens to be located in these internal central processor scratchpad registers which software cannot access.

Prior Art:

The problem has been solved in one of two ways. The first method is to simply re-initialize the entire system, loading the system base with different contents as a result of a new initialization process. This technique is extremely time-consuming and quite undesirable.

A second technique for solving the problem is to have the software employ special model-dependent properties of the particular machine on which it was running, to change the system base. This results in the very undesirable property of completely model-dependent software—a feature which certainly does not serve to endear a computer manufacturer to a customer.

What is needed therefore is a technique of changing information in the system base both in main memory and internal scratchpad registers making the system base in scratchpad memory accessible to software but not visible to software i.e. cannot be read by the software.

OBJECTS OF THE INVENTION

It is a primary object of the invention to provide an improved system and method for changing information in the system base of a computer system.

It is another object of the invention to provide a switch-system base instruction in firmware which changes, upon execution, information in the system base.

It is still another object of the invention to provide a system and method for changing information in the system base which is located in main memory and scratchpad registers.

Yet another object of the invention is to provide an improved switch system base mechanism.

These and other objects of the invention will become apparent from the description of a preferred embodiment of the invention when read in conjunction with the drawings contained herewith.

SUMMARY OF THE INVENTION

A model-independent instruction in firmware is provided to change, upon its execution, information in the system base which is located in both main memory and internal scratch-pad registers.

The system base located at a location specified by the boundary address register BAR in main memory, is copied into a new location in main memory with predetermined portions of its information changed. This is readily accomplished by the operating system. (See application incorporated by reference into the application herein). The switch system base instruction then transfers the information which is in the system base at the new location into the system base at the location specified by BAR and also that portion of the system base located in scratchpad memory.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features which are characteristic of the invention are set forth with particularity in the appended claims. The invention itself, however, both as to organization and operation together with further objects and advantages thereof may best be understood by references to the following description taken in conjunction with the drawings in which:

Figure 1 is a block diagram of a multiprogramming system utilizing the invention.

Figure 2 is a schematic representation of various hardware structures utilized by the invention.

Figure 3 is a legend of terms used for reserved areas of storage in registers depicted in Figure 2.

Figure 4 is a schematic diagram of a process control block.

Figure 5 is a schematic diagram of a system for addressing a process control block.

Figure 6 is a schematic diagram of the system base of the invention.

Figure 7A and 7B are a schematic representation of a stack segment and a stack frame respectively.

Figure 8 is a schematic diagram of a system for addressing G-segments and in particular the queue of processes in the GO segment.

Figure 9 is an exploded schematic diagram of a GO segment illustrating queue of processes and process linking.

Figures 10a through 10L are block diagrams of structures in the PCB.

Figures 11a through 11r are block diagrams of structures in the system base.

Figure 12 is a schematic diagram of addressing schemes of user and system segments utilizing the system base and PCB structures.

Figure 13 is a schematic diagram of the control unit.

Figures 14a through 14i are flow diagrams of the dispatcher unit in firmware.

Figure 15a is a structure of the switch system base instruction.

Figures 15b-15c are flow diagrams of the switch system base instruction in firmware.

DESCRIPTION OF A PREFERRED EMBODIMENT

GENERAL DISCUSSION

The invention operates typically in the hardware system environment hereinafter described coordinated by a hardware/firmware/software operating system. Referring to Figure 1 the subsystems are the processor subsystem 101, the storage subsystem 102, and one or more — up to 32 — peripheral subsystems 103. The processor subsystem contains a central processing unit (CPU) 104 and up to four input/output control units (IOC) 105. Each peripheral subsystem consists of a peripheral control unit (PCU) 106, a number of device adapters (DA) 107, and up to 256 peripheral i/o devices 108. The storage subsystem 102 consists of one to four semiconductor memory modules of 32 to 512 kilobytes each.

I. PROCESSOR SUBSYSTEM

In the processor subsystem 101, the CPU 104 performs the basic processing operations for the system, and interfaces with memory 102. The IOC 105 controls all information exchanges between the storage subsystem 102 and peripheral devices 106.

A. CENTRAL PROCESSING UNIT

The CPU includes a main memory synchronizer 109, a buffer store 110, various elements that comprise the computational unit 111, and optional emulation facilities 112. The main memory synchronizer 109 resolves conflicts for the use of main memory among the computational unit

111, the buffer store 110, and the IOC 109. Conflicts are resolved on a priority basis: the IOC has the highest priority followed by memory writes (from the computational unit) and then memory reads (into the buffer store). The main CPU also includes the address control unit ACU 131 which controls main memory addressing and the associative memory AS 132 used to store most recently used addresses of main memory. The buffer store 110 is a small high-speed buffer memory that reproduces a selected region of main memory and interfaces with the computational unit to decrease average memory access time. During each memory read, both the buffer store and main memory are accessed. If the information to be fetched is already in the buffer store, the main memory read is terminated and the information fetched from the buffer store. Otherwise the main memory 102 is read. Every time this is done, the CPU 104 fetches 32 bytes that contains the desired information. This information remains in the buffer store for future memory references. Since the buffer store is transparent to software, the program controlling the computer at any given moment cannot determine whether the information it is processing has been fetched from the buffer store or from main memory.

The computational unit 111 performs all data processing and address generation within the CPU. A typical control store 130 within the computational unit (see a book entitled Microprogramming: Principles and Practices, Samir S. Husson, Prentice Hall, Inc.) contains firmware which initializes the system, controls the CPU 104 and IOC 105, and decodes an instruction set (not shown). Optionally the control store may provide scientific instructions, test routines, emulation packages, or special purpose features which extend the capabilities of the processor subsystem.

As an option, the CPU provides emulation of systems other than the instant system. Emulators 112 are com-

ponents of firmware, software, and in some instances hardware.

B. INPUT-OUTPUT CONTROL UNIT

The IOC 105 portion of the processor subsystem provides a data path between any peripheral subsystem 103 and the storage subsystem 102. This path allows for the initiation of peripheral commands and controls the resulting data transfers. An IOC can typically handle up to 32 channel control units (not shown).

C. PERIPHERAL SUBSYSTEMS

In a peripheral subsystem 103 on Figure 1 the PCU 106 is a stand-alone microprogramming processor that relieves the load on the CPU 104 by controlling the i/o devices 108 during i/o operations. The PCU does this by executing instructions contained in a channel program. This program results in arithmetic, logical, transfer, shift, and branch operations being performed in the PCU. There are several kinds of PCU's according to the kind of device each controls: i.e. unit record, mass (disk) storage, magnetic tape, communications, etc.

Device adapters 107 mediate between every PCU and the devices it controls. Each contains the dedicated firmware and logic necessary to implement communication with a particular type of device. Depending on the type, a DA 107 controls one or several devices.

The major functions performed by a peripheral subsystem 103 are as follows:

1. Transforming CPU instructions into a series of commands acceptable to the appropriate peripheral device.
2. Packing and unpacking data in the form needed by the CPU or the appropriate peripheral device.

3. Keeping the CPU informed of the status of the subsystem and of the devices under its control.
4. Independently initiating and processing error and recovery procedures.
5. Allowing on-line diagnosis of a device without disturbing the device-sharing capabilities of the associated peripheral processor.

A PCU resolves conflicts for main memory between devices attached to it; however, the IOC resolves conflicts between PCU's.

D. STORAGE SUBSYSTEM

Each memory module 1-4 is 4 or 8 bytes wide. The number of modules, their size, and the data path width may vary according to size of computer. Memory modules are four-way interleaved in such a way that the four modules are accessed sequentially (module 1 contains the first 8 bytes, module 2 contains the second 8 bytes, etc.). Interleaving decreases the number of conflicts for access to main memory and thereby decreases the average memory access time. Memory is reconfigurable in case of failure; i.e., blocks of memory within a module may be removed without destroying contiguous addressing.

Main memory 102 consists of a capacitive storage medium in the form of metal oxide semiconductor (MOS) chips. This medium operates on the refresh principle to maintain information. Each memory location is typically refreshed at least once every 2 milliseconds; the design ensures that few conflicts occur between refresh timing and memory accesses. (In cases of conflict, refreshing takes precedence).

An area at the beginning of main memory is reserved for hardware and firmware. The upper limit of this area is

defined by the content of a boundary address register (BAR - to be later described) which is visible to the system software. The BAR content is set at system initialization time. The memory area below the address specified in the BAR can contain IOC tables which define the configuration of the peripheral subsystems, firmware to control the CPU, or microprograms and tables for emulation. The size of the area below the address specified in the BAR depends on the system configuration. Whether microprograms are in main memory or control store depends on the system configuration and the applications run on the system.

II. BASIC MACHINE STRUCTURES

There are typically three basic data structures utilized in this hardware: data formats, software visible registers, and the instruction formats.

A. DATA FORMATS

Information is transferred between memory and the CPU in multiples of 8 parallel bits. Each 8-bit unit of information is called a byte. Parity or error correction data is also transferred with data but cannot be affected by software. Therefore, in this patent specification the term data excludes the associated parity or error correction data.

B. BYTES

Bits within a byte are numbered 0 through 7 from left to right. Bytes are processed separately or in groups. Two bytes constitute a halfword, 4 bytes a word, 8 bytes a doubleword, and 16 bytes a quadword. These are the basic formats for all data, including instructions.

C. DATA REPRESENTATION

All data are in binary form, but may be interpreted as binary, decimal, or alphanumeric. Data bits are interpreted in groups of four, as binary coded decimal data; eight as alphanumeric, or 16 to 64 as binary digits. The latter are interpreted as signed, fixed, or floating-point numbers in binary notation. Any number of contiguous bits up to a doubleword may also be manipulated as a string. The alphanumeric character set is represented in EBCDIC. ASCII is supported as an alternate exchange code.

D. BYTE ADDRESSES

Byte locations in main memory are consecutively numbered starting with zero; each number is the address of the byte. A group of consecutive bytes is said to be halfword-, word-, doubleword-, or quadword-aligned, if the address of the left byte in a group is a multiple of 2, 4, 8, or 16, respectively. Whenever a halfword, word, doubleword, or quadword is so aligned, that unit can be fetched from that address. The location of data in main memory is specified by a data descriptor which is accessed indirectly during address development. (See Patent Application No. _____ filed _____ entitled Segmented Address Development and assigned to the same assignee as the instant application).

E. VISIBLE REGISTERS

There are 33 user-visible registers in the CPU 104 Figure 1 whose contents collectively define the state of the CPU. There are four types: (See Figure 2).

1. general registers

2. base registers
3. scientific registers (optional)
4. miscellaneous registers

F. GENERAL REGISTERS

General registers (GR) 201 are used to manipulate fixed-point binary numbers and bit strings. There are typically sixteen 32-bit general registers in the CPU 104—GR0 through GR15. General register GR8 through GR15 are also usable as index registers. When used as index registers, they are herein called X0 through X7: Indexing is performed using the 32-bit two's complement integer contained in a register.

G. BASE REGISTERS

Base registers (BR) have the same format as instruction counters IC and stack registers 202-203. Base registers are used during address computation to define a part of memory. There are typically eight 32-bit base registers, BR0 through BR7.

H. SCIENTIFIC REGISTERS

Scientific registers (SR) are optional equipment for computation with floating-point binary numbers. There are typically four 8-byte scientific registers which are referred to as SR0 through SR3. Scientific registers have the format 204-205 of Figure 2.

I. MISCELLANEOUS REGISTERS

There are five other registers:

- instruction counter - having format 202-203;

- status register - having format 207;
- stack register (called the T register);
- boundary address register - having format 202-203; and
- hardware control mask register - having format 208.

The instruction counter (IC) is a 32-bit register that contains the address of the instruction being executed. The status register (STR) 207 is an 8-bit register that records facts about the procedure currently being executed, for example, whether an underflow was caused by the most recent operation. The stack register also known as the T-register is a 32-bit register that contains a pointer to the top of a pushdown stack associated with the currently active procedure. Stacks to be described infra provide a work space, and a mechanism for saving local variables and preserving procedure entry, and return information. The boundary address register (BAR) 206 is a 28-bit register which specifies the lowest absolute main memory address accessible by software. This register is loaded during system initialization and can only be read by software. The hardware control mask register 208 is an 8-bit register which records machine condition information.

J. INSTRUCTION FORMATS

There are approximately 200 instructions although more or less may be utilized. Each instruction is one of four different lengths but always an even number of bytes long. Instructions are stored in consecutive storage locations. The address of the leftmost byte is a multiple of 2, and is the address of the instruction.

The eight most significant bits (and in some cases bits 8 through 11 or 12 through 15) of an instruction represent the operation code, while the remaining bits represent one or

more operands. An operand may be a register designator, displacement designator, address syllable (logical address) literal value, immediate literal value. The type and number of operands are determined by the instruction format.

III. SYSTEM ORGANIZATION

A. JOB STEP AND TASK

Work to be performed by the computer system is defined externally by a series of job steps via a job control language. A job step is a unit of work to which hardware resources are allocated. Typically a job step consists of several tasks. A task is the smallest unit of user defined work consisting of a stream of instructions executed without parallelism.

B. PROCESS

The user-visible concepts of task and job step are represented in the hardware by a process and process group, respectively. A process is defined as an ordered sequence of instructions which can be executed asynchronously by the CPU (i.e., several processes can be active and sharing resources, but only one process is actually running at any one instant). A process group is a related set of processes necessary to perform one job step.

C. PROCESS CONTROL BLOCK AND SYSTEM BASE

Because processes can relinquish CPU control at various points during their execution, a storage area in main memory is made available to a process to save CPU status. This status information is utilized to precondition the CPU before a process regains control of the CPU.

The storage area assigned to a process is called a process control block (PCB) 400 on Figure 4. The data contained in a PCB include the addresses of memory areas (address space) assigned to the process, the contents of all pertinent registers, and the state of the process. Thus a PCB serves as a temporary storage area for information necessary to start or restart a process without any information loss. Each PCB is visible to the hardware and can be addressed by the operating system via a set of hardware tables developed during system initialization and modified during system operation (Figure 5).

There is an absolute main memory area which is referred to as the system base (Figures 5 and 6). This area is developed by firmware and is accessible via the base address register (BAR) 501 which can be read but not written. The system base 502 contains a number of system attributes which include a job step number and a process group number (J, P) respectively for the currently running process. Another attribute in the system base is a pointer to a hardware defined data structure known as the J table 503. This table contains an entry for every job step presently in the system. Each entry in the J table 503 points to an associated P table 504 which is also a hardware defined data structure. This table defines a process group and contains an entry for every process in the process group. Each P-table entry points to a PCB 400.

Referring to Figure 5 the J-table pointer 505 indexed by the J number via computational unit 506 provides access to a J-table entry 503. This entry contains a P-table pointer which when indexed by the P number via computational unit 506 provides access to a P-table entry 504. The P-table entry contains a pointer 507 to the PCB of the current running process. Thus the operating system can access the active PCB using the contents of the BAR 501 and can access any other PCB given its associated (J, P) logic name.

D. MEMORY SEGMENTATION

In a multiprocess environment, such as herein described there are many processes in memory at any given time. These processes vary in size and demand for memory which causes a memory allocated problem. The hardware herein described in cooperation with an operating system (not shown herein) solves the problem by dynamically allocating memory space. Due to the random nature of memory requirements, memory is allocated in variable size segments and the memory allocation can be restructured during process run time. Thus, a process may be allocated a number of noncontiguous memory segments. This memory allocation method is called segmentation.

Segmentation presents an additional problem in that memory addresses have to be modified whenever part or all of a process is relocated. To alleviate this problem the system herein described provides a technique whereby addresses used by a process are logical rather than absolute main memory addresses. These logical addresses are used to develop absolute addresses.

Segmentation also allows each process to access its own or related memory segments via a system of segment descriptors. By accessing a segment descriptor, a process can obtain the address of a segment. Segment descriptors are contained in main memory and are maintained by the operating system.

Each process may have access up to 2068 memory segments. Normally, this would require an equal number of segment descriptors per process. However, since segments can be shared, the operating system groups segment descriptors into segment tables. This grouping is based on accessability by one process (task), a process group (job step), or globally (system wide). Each process may have up

to 15 segment tables associated with it. This technique requires only one segment descriptor for each segment which can be accessed by a process via a segment table. Thus, the memory space required for segment descriptors is decreased; memory updating during relocation is reduced; and some program protection is provided. (The main mechanism for program protection is the ring system. See Patent Application No. _____ filed _____ entitled "Protection of Information in a Multiprogram, Multiprocessor Computer System;" also see Patent Application No. _____ filed _____ entitled "Ring Crossing Mechanism" both assigned to the same assignee as the instant invention.

A process must be able to determine which segments it is allowed to access. Accordingly, the system provides a process with two segment table word arrays (STWA). These arrays contain the addresses of all segment tables accessible to a process. There are two segment table word arrays per process because there are two segment sizes, large and small. Large segments have a maximum size of 2^{22} bytes while small segments have a maximum size of 2^{16} bytes. All segments vary in size in 16-byte increments up to the maximum. A system can typically accomodate up to 28 large segments and 2040 small segments.

Segment table word arrays may be relocated by the operating system; therefore, a process must know the absolute address of its associated STWA's. The PCB for any process contains two words which contain this information which are known as address space words ASW0-1 on Figure 4. Each word points to a segment table word array STWA. The operating system updates the contents of the ASW's whenever the associated STWA's are relocated. Working down the chain of pointers and decoding the segment descriptor is a firmware function and thus once initiated is not visible even to the operating system.

Segmentation defines over 200 million bytes of address space as being available for processes. This number exceeds the capacity of main memory; therefore, a secondary storage (magnetic disk or drum) is used in conjunction with main memory. The operating system creates the illusion that the system has a much larger main memory than is really available. This concept is called virtual memory.

At any given time, a defined segment may or may not be physically in main memory. The contents of a segment descriptor indicates whether or not the associated segment is in main memory. The hardware detects any attempts by a process to access a segment not in main memory and notifies the operating system. The operating system causes the desired segment to be loaded into main memory from secondary storage. Then the operating system places the segment's memory address in the segment descriptor which is the only place where the absolute address of a segment can be found. This operation is invisible to the process and thus it is not aware that the segment was not in main memory or that it may have to be relocated in main memory. (For details on memory segmentation see Patent Application No. _____ filed _____ and entitled "Segmented Address Development". Also see patent No. _____ .)

The computer system herein described provides data and procedure protection by preventing processes from interferring with each other or sharing each other's address space in an unauthorized manner. This protection is accomplished by restricting addressability via memory segmentation and by a ring system.

The segment tables isolate the address space of the various processes in the system. Processes always use a segmented address during execution. A segmented address consists of a segment number and a relative address within the segment (see above referenced application on Segmented Address

Deve' ›pment). The hardware checks that the address used by a process is part of the address space assigned to the process. If the address is outside the prescribed address space, an exception occurs. A process cannot refer to data within the address space of another process because the hardware uses the segment tables of the referencing process. Thus, there is no possibility for a process or process group to reference an entity belonging to another process group.

Generally, overlap in address space in the system occurs for those segments shared by all processes. These public segments are created by system programs which check to insure against address conflicts. Thus, segmentation protects user programs against each other and protects the operating system against user programs.

Segments shared by several processes are not protected from misuse by one of these processes. To solve this problem, a ring system is utilized whereby procedure and data segments are grouped into a four-class hierarchy. The four ring classes are numbered 0 through 3. Each ring represents a level of system privilege with level 0 (the innermost ring) having the most privilege and level 3 (the outermost ring) the least. Every procedure in the system has a minimum and a maximum execute ring number assigned to it which specifies who may call the procedure. A procedure is a subroutine which is capable of calling other procedures and passing parameters to them:

The general rules of the ring system are as follows:

1. A procedure in an inner ring has free access to data in an outer ring. Conversely a procedure in an outer ring cannot access data in an inner ring.
2. A procedure in an outer ring can branch to a procedure in an inner ring, but the reverse is not allowed.
3. Each segment containing data is assigned two ring values, one for read (RD) and one for write (WR).

These ring values specify the maximum ring value in which a procedure may execute when accessing the data in either the read or write mode.

Each time a procedure instruction is executed, the procedure's ring number (effective address ring, EAR) is checked against the ring numbers assigned to the segment containing the referenced data. The EAR is the maximum number of process ring numbers in the instruction counter and all ring numbers in base registers and data descriptors found in the addressing path. Access to the data will be granted or denied based on a comparison of the ring numbers. For example, if a system table exists in a segment having a maximum read ring value of 3 and a maximum write ring of 1, then a user procedure executing in ring 3 may read the table but may not update the table.

By predesign, rings 0 and 1 are reserved for the operating system and rings 2 and 3 are reserved for the user. Ring 0 contains those segments critical to total system operation. Ring 1 contains the bulk of the system segments whose failure would not be catastrophic and would allow recovery. The user may utilize ring 2 for checked-out programs and ring 3 for programs being debugged.

F. PROCEDURE CALLS

The procedure call is an important function in the system herein described. Procedure calls are used to pass from one procedure to another; to allow user procedures to employ operating system services; and to achieve a modular structure within the operating system. A procedure call is effected by instructions and a hardware recognized entity called a stack (Figure 7A).

A stack is a mechanism that accepts, stores and allows retrieval of data on a last-in-first-out basis. Stacks reside in

special segments called stack segments. A stack segment consists of a number of contiguous parts called stack frames 701 (Figures 7A and 7B) which are dynamically allocated to each procedure. The first stack frame is loaded into the top of the segment and succeeding frames are loaded after it. The last frame loaded is considered the top of the stack. The T-register 702 locates the top of the stack for the currently active process. A virtual T-register exists in the PCB of all other processes in the system.

A stack frame 701 of Figure 7B consists of three areas: a work area 702 in which to store variables, a save area 703 in which to save the contents of registers, and a communications area 704 in which to pass parameters between procedures. Prior to a procedure call, the user must specify those registers he wishes saved and he must load into the communications area the parameters to be passed to the called procedure. When the call is made, the hardware saves the contents of the instruction counter IC and specified base registers to facilitate a return from the called procedure.

Each procedure call creates a stack frame within a stack segment 701 and subsequent nested calls create additional frames. Each exit from one of these called procedures causes a stack frame to be deleted from the stack. Thus, a history of calls is maintained which facilitates orderly returns.

To insure protection between procedures executing in different rings, different stack segments are used. There is one stack segment corresponding to each protection ring per process. A PCB contains three stack base words which point to the start of the stack segments for rings 0, 1 and 2 associated with the process. The ring 3 stack segment can never be entered by an inward call; therefore its stack starting address is not required in the PCB.

IV. PROCESS MANAGEMENT AND SYNCHRONIZATION

The system herein provides for multiprocessing operations which are controlled by an operating system using a combination of software, hardware and firmware. Software creates and deletes processes within the system while hardware and firmware multiplex processes on the CPU. In addition, a combination of software, hardware and firmware provide for synchronization between processes.

Processes are normally, but not always, started and stopped at the initiation and termination of i/o operations, during related job handling, and at other times for purposes deemed necessary by the operating system. Therefore, a communications system is necessary to efficiently start and stop related processes and to pass information between them. The hardware system herein provides internal messages called semaphores to provide a communications link between the processes.

A. PROCESS STATES

A process can be in one of four possible states at any time: running, ready, waiting or suspended. The hardware recognizes these four possible process states and executes various firmware procedures to effect process dispatching, state changes and to maintain data structures based on a process's state. The PCB contains a state field which defines the current state of its associated process.

A process is in the running state when it has control of the CPU. This state involves supplying the CPU with an address space (segment tables) and a starting address. The CPU then executes instructions in the procedure segments of the process. The process name J table word (logical

address) of the PCB for the currently running process is retained in the running process word (BAR +60) within the system base (Figure 6). (Note: The system base shown in Fig. 5 is the same as that shown in Fig. 6, but with some details omitted.)

The ready state is equivalent to running state except that the process does not have control of the CPU because it has not been recognized by the CPU. A process in the ready state is in contention for the CPU with other ready processes and the running process.

A process is in the wait state when it cannot continue until a specific event occurs such as a message via a semaphore. A waiting process is not in contention for the CPU but it may be in contention with other waiting processes for the required event.

A suspended process is a process which has been stopped for a time by software and may be resumed later. The decision to stop and resume the process is external to the process. Thus, a suspended process is not active and therefore cannot receive notification of event occurrences and cannot utilize the CPU.

A process is suspended under the following conditions:

(1) By executing a Terminate instruction (as a result of having completed all its functions.)

(2) By execution of a Suspend instruction by the operating system.

(3) By the occurrence of an exception condition whereby control is transferred to the operating system.

B. PROCESS DISPATCHING

Processes move from one state to another voluntarily by action of the process while running or involuntarily by the actions of other processes. CPU firmware, known as the

dispatcher, controls the transaction of processes between states. The dispatcher uses a set of queues (to be later described) to manipulate processes which are in the ready or the waiting states. Suspended processes are controlled by software.

Referring to Figures 6, 8 and 9, a ready or waiting process is represented by a PCB and a special queue entry called a process link. Figure 9 shows an exploded view of contents of the GO segment 802, and contains process links 803a-803b and 803c-803g of active processes, and free process links 805a-805c of suspended processes. Each process link specifies the process name (J, P), the process priority and a pointer to the next process link in the queue. There are various types of queues such as wait queue 803a-b and ready queue 803c-g.

A hardware device similar to the J table, known as the G table, (Figures 6 and 8) contains pointers to all general (known system wide) segments 802-802n. The first element, GO, of the G table 801 points to that segment 802 containing the dispatcher queues. A G-table pointer to the G table 801 is found in the system base 502 on Figure 5. Also in the system base is an entry called the internal process queue word (IPQW) which identifies the head 805 of the ready queue 803c-803g in the GO segment 802.

Thus, the dispatcher can examine all ready processes by consulting the ready queue 803c-803g. When the currently running process changes states, the dispatcher removes the process link at the head of the ready queue and uses the J, P name to access its PCB. The process defined by the PCB then becomes the new running process.

Since more than one process may be awaiting on the same event, a queue of waiting processes 803a-803b exists for each event. Waiting processes are also strung together via process links 805 residing in the GO segment. A pointer

to the head of a wait queue exists in a semaphore 903 (to be later described). A number of events exist for which a process may wait; therefore, there are a number of wait queues each of which has an associated semaphore 903, 904.

The number of processes ready or waiting varies dynamically. Thus, the number of process links required for the ready and wait queues also varies. This fact introduces a memory management problem for the dispatcher. The problem is solved by another queue called the free process link queue 805a-c. This queue links together all process links in segment GO that are not being used by the ready or the wait queues and can be used to extend a particular queue of ready or waiting processes. A pointer 901 to the head 902 of the free process link queue 805 resides near the beginning of the GO segment 802.

C. PROCESS SYNCHRONIZATION

Process synchronization is required to coordinate the activities of two processes working on the same task. The synchronization is achieved using semaphores 903-904 which are data structures residing in the address space of communicating processes. A semaphore is used to signal event occurrence and to handle queues of messages. An event in this context is anything observed by a process which may be of interest to some other process. The event may be the completion of an asynchronous operation or the availability of a resource.

A process uses two semaphore operations to signal an event occurrrence. One operation sends a signal to a semaphore; the other picks up a signal from a semaphore. (The sending operation is often called a V-operation; the receiving operation is called a P-operation). The sending

operation allows a process to send data or a signal that data are ready. The semaphore stores the signal until another process is ready to pick it up. Thus, the sending process is free to proceed, since it has sent the data. The receiving operation examines a specified semaphore and picks up the signal. If a signal is present, the receiving process continues executing. However, if there is no signal at the semaphore, the receiving process enters the wait state. The semaphore then serves as a pointer to the head of a wait queue. The process remains in the wait state queued at the semaphore until another process sends a signal to that particular semaphore. Thus, a semaphore can hold a signal until a process picks it up, or a semaphore can hold a process until a signal is sent to it.

Messages can also be passed from process to process. A message has the same present or not present quality as a signal plus additional information. Part of the information is supplied by hardware and part is supplied by the procedure of the process that sent the message. A message carries the process name of the sending process. Thus, many processes can send information through a single semaphore stamped with the sender's name.

A message semaphore may have a queue of messages waiting to be picked up by processes. As with signal semaphores, requirements for memory space increases and decreases thus presenting a memory management problem. Again, the problem is solved with a queue of free message links. These links reside in a known place in a segment that can easily be found when needed to supply or absorb message links.

Because semaphores and the queues built on them are shared by different processes, the total semaphore structure is protected. This is accomplished by hardware and software conventions that restrict access to any segment containing

semaphores. Thus, semaphores must be in semaphore descriptor segments, some of which may be G segments (if system communications is necessary). However, all G segments (except GO) are semaphore descriptor segments.

Each semaphore descriptor contains a pointer to a semaphore. Semaphore addresses are developed via a semaphore descriptor, thus providing added protection for the semaphore. A semaphore segment can be addressed logically using a segment number and a relative location within the segment or directly using the G, D number.

PROCESS CONTROL BLOCK STRUCTURES

Referring to Figure 4 there is shown the format of the process control block (PCB). The process control block 400 is a storage area in main memory made available to a process to save the CPU status. Addressing a PCB is performed as described supra in relation with Figure 5. The PCB pointer 507 (Figure 5) points to the process control block PCB at memory location 0 on Figure 4. It will be noted that proceeding in a downward direction memory locations increase by 4 bytes whereas in proceeding in an upward direction from memory location 0 they increase by 8 bytes. The downward memory locations are considered positive from 0 whereas the locations in an upward direction from 0 are considered negative directions. The upward locations are optional and may or may not be included in the process control block; also locations 148 through 176 are also optional. (Note that the numerals under memory location specify the displacement in bytes from the 0 reference location of the process control block PCB and are not to be confused with the reference numerals commonly used to identify parts in a patent drawing). Starting at byte 0 up to but not including byte 16 there are stored four process main

words PMW 0 through PMW 3 with each process main word PMW being four bytes in length. Process main word 0 occupies bytes 0 through 3 and is comprised of 4 parts: a capability byte, a priority byte, a state byte and a decor extension byte DEXT. Referring to Figures 10a through 10d there are shown details of process main word PMW 0, with further details of the capability byte 1001 shown on Figure 10b. Referring to Figure 10b, the first bit 1005 is the accounting mode bit for indicating whether or not time accounting functions are being performed for the process. When the accounting mode bit 1005 is set to binary 0 no time accounting function is being performed for the process; whereas when the accounting mode 1005 is set to binary 1, time accounting is being performed. The scientific mode bit 1006, when set to zero, indicates that saving of scientific register of the machine is not performed and the scientific register saving area located at bytes 148 to 176 on Figure 4 does not exist in the process control block PCB. When the scientific mode bit 1006, is set to binary 1, the scientific optional feature exists and is being used in the process, and the scientific registers saving area is used to save the contents of the scientific registers when necessary. The code mode bit 1007 indicates whether or not a standard code set or compatibility code set is being used by the process, with a binary 0 in that position indicating that standard code set is being used; whereas a binary 1 in the third bit position 1007 indicates a compatibility code set is being used. The remaining of the bits of the capability byte are set to zero.

Details of the priority byte 1002 are shown on Figure 10c. Referring to Figure 10c the first four bits 1008 of priority byte 1002 is utilized to set the priority level of the process associated with that given process control block PCB. Each process is assigned one of 16 levels of priority which is used for ordering competing processes i.e. (a) for choosing the

process to be run among ready processes, (b) for putting processes in queues. Priorities decrease from 0 to 15, and for a given priority level the FIFO (first in first out) rule is applied. The next 4 bits 1009 of priority byte 1002 are zeroes.

Referring to Figure 10d details of the state byte 1003 are shown. A state byte is utilized to provide information with regard to the process associated with the process control block PCB 400. The active field bit A 1010 is set to binary 1 when the process is activated. The suspend field S 1011 is set to binary 1 when the process is suspended. The substate field SS 1012 is a 2 bit field and defines the following substates of the process: (a) when set to binary 00 the process is inactive; (b) when set to binary 01 the process is waiting in the queue of ready process (Q/PR/RDY); (c) when set to binary 10 the process is waiting on a semaphore in a queue of semaphores (Q/PR/S); (d) when set to binary 11 the process is being executed by the processor. The mid-operation field (MOI) 1013 is set to binary 1 when an interrupt happens and is taken care of during the execution of an instruction—i.e. before the completion of the process. The extended decor mode bit EXTD 1014 is set to 1 when the process is operated in an extended decor mode which is an emulation mode of the machine. Bits 1015 and 1016 are set to 0. The fourth byte of process main word PMW 0 contains the decor extension number and is utilized when the system is in emulation mode.

Process main word PMW 1 is stored in bytes 4-7 of the process control block PCB. Details of PMW 1 is shown on Figure 10e. The status byte 1016 is the first byte in PMW 1 and stores the status register contents. The multiprocessor byte MP 1018 is significant in a multiprocessor architecture otherwise this field is zero. The second and fourth bytes of process main word 1 are the MBZ fields 1017 and 1019 respectively which must be zero for normal operation.

Process main word PMW 2 occupies bytes 8 through 11 of the process control block and is shown in more detail on Figure 10f. Referring to Figure 10f the field from bit 4 through bit 31 contains the local name SEG, SRA 1021 of the semaphore to which the PCB is linked when the process is either in the waiting or suspended states. The exception class and type field 1023 contains the class and the type of the interrupt-like exception which cause the process to enter the suspended state after an exception. The field from bits 4 through 15 is meaningless 1022 when a process is in a different state than those mentioned above.

Process main word PMW 3 occupies bytes 12 through 15 in PCB 400 and points to a decor extension table. Referring to Figure 10g for details of PMW 3 the DETSZ field 1024 defines the number of entries in the table and if this field is zero no decor extension is allowed to the process. The DETA field 1025 is the absolute address of the decor extension table in units of 16 bytes and is significant only if DETSZ is not 0. The decor extension table is made up of DETSZ entries. Each entry is one byte size. The DEXTth entry of the table defines the capability of the process to operate in the decor extension mode DEXT. When the DEXTth byte is 0 the decor extension number DEXT is not allowed, whereas if the DEXTth byte is 1 the decor extension number DEXT is allowed. Values of DEXT other than 0 and 1 are illegal. (See Figures 10a DEXT number 1004).

Bytes 16 through 23 of PCB 400 contains 2 address space words ASW 0 and ASW 1 respectively and each ASW contains a pointer to an array of segment table words. Both ASW 0 and ASW 1 respectively have the same format shown on Figure 10h. The size of the array of the segment table words is defined by the number of segment table words in an array and typically comprises six for ASW 0 and eight

for ASW 1. The STWSZ field 1026 indicates the size of the array of the segment table words. The segment table word array field STWA 1027 contains the absolute address STWA of the array in units of 16 bytes—i.e. the absolute address of the array is 16 times STWA in bytes.

Bytes 24 through 27 in the PCB contain an exception word EXW shown in greater detail on Figure 10i. The exception word contains a pointer (SEG, SRA) 1029 to an exception class table which defines the action to be taken following a process exception according to its class as stored in process main word PMW 2. (See Figure 10f). The MBZ field 1028 of exception word EXW must be 0.

The stack word SKW located in bytes 28 through 31 of the PCB contains the value of the top of the T register of the stack of the process when the process is not running and is shown in greater detail in Figure 10j. Referring to Figure 10j, bits 0 and 1 define the TAG field 1030. The TAG indicates the type of descriptor by its contents and must be zero for SKW. Bits 2 and 3 of the SKW word contain the RING field 1031 which contains the ring number associated with the segmented address of the stack for protection purposes and in this case must be zero. Bits 4 through 31 contain the segment number SEG, and the segment relative address SRA 1032 and is a field which identifies the segment described in a segment table and the segment relative address within the segment. The stack word SKW is updated every time the process leaves the running state. It is used to restore the T register contents every time the process becomes running. In this last case the TAG 1030 and RING 1031 are tested to be zero, otherwise an illegal PCB exception occurs.

Bytes 32 through 35 of the PCB 400 contain the instruction counter content word ICW sometimes also referred to as ICC. Referring to Figure 10k there are shown details of the instruction counter word ICW wherein the TAG field

1033 must contain binary 00 (i.e. values other than zero are illegal in the instruction counter). The current RING field 1034 occupying bits 2 and 3 defines the current ring number of the process to be used in determination of access rights to main storage. Bits 4 through 31 define the segment number and the segment relative address (SEG, SRA) 1035 which define the address of the next instruction to be executed.

The MBZ field in bytes 36 through 39 must be zero. (Note the MBZ field always indicates a field which must be zero). The MBZ word is tested every time the PCB is accessed from the name J, P. If it is not zero an illegal PCB exception occurs.

Stack base words SBW 0-2 occupy bytes 40-51 in the process control block 400. These words have the same format which is shown in greater detail on Figure 10 l. They are utilized during stack operations and whenever used their TAG field 1036 and RING field 1037 must be zero otherwise an illegal PCB exception occurs. Bits 4 through 31 contain the segmented address (SEG, SRA) 1038 of the first bytes of the stack segments for ring zero, 1 and 2 respectively.

Bytes 52 through 83 of the process control block 400 is a space reserved for the base registers saving area (8 words). Bytes 84 through 147 is a saving area which is utilized to save the values of all general registers (16 words). Bytes 148 through 179 is a saving area which is utilized to save the scientific registers (8 words).

Five double words are provided in the PCB 400 above the PCB zero address, for time accounting purposes when the accounting mode bit in the PMW 0 word is set. These words are located from PCB address minus 8 to PCB address minus 40. Each word contains a time or a time interval expressed in microsecond units in its first 52 bits with bits 52-63 filled with zeroes. The residual time out double word RTO (first 8

bytes above 0 in the PCB) contains the quantum of time which is actually spent by the processor on behalf of the process before a time out exception occurs. The RTO word is updated in the following way: each time the process exits the running state the process timer value is stored in the RTO word. Each time the process enters the running state, the process timer value is loaded from the RTO.

The running time accounting RUA double word at bytes 7 through 15 is a time counter which specifies the total amount of processor time a process was in the running state. The time accounted for is the time actually spent by the processor on behalf of the process exclusively. The RUA word is updated in the following way: each time the process exits the running state, the value of the process timer PT is read. The difference of the contents of RTO and PT is added to RUA. (Consecutively, the PT value is stored in RTO). Note that the time during which the process is suspended is not computed. The RTO and RUA words are updated even if the accounting mode bit is set to 0. However the CET, RTA, and WTA words (to be later described) are provided in the process control block only if the accounting mode bit in the process main word PMW 0 is set to 1. They are updated only in this case.

The waiting time accounting WTA word at bytes 17 through 23 is a real time counter which specifies the total amount of real time the process was in the waiting state. The WTA word is updated in the following way: each time the process exists the waiting state the time of day clock (not shown) value TOD is read and the Value of TOD minus the value of CET word is added to the WTA word.

The ready time accounting RTA word located at bytes 24 through 31 is a double word which is a real time counter which specifies the total amount of real time the process was in the ready state. The RTA is updated in the following way:

each time the process exits the ready state, the time of day clock value TOD is read, and the contents of TOD minus the contents of CET is added to RTA.

The current entry time CET double word at bytes 32 through 39 contains the time of day at which the process entered one of the following states: ready, waiting, running, and suspended.

SYSTEM BASE STRUCTURES:

Referring to Figure 6 the format of the system base 600 is shown. The system base resides in absolute main memory and is developed by firmware and is accessible via the boundary address register (BAR) which can be read but not written. The boundary address register BAR is below an area in main memory reserved for hardware and separates this area in memory reserved for hardware and the system base 600. Referring now to Figure 6 the system base 600 contains a number of system attributes which includes a job step number and a process group number (J, P) for the currently running process. From the logical name of the process J, P, the absolute address of the corresponding process control block PCB is obtained. The size and address of the J table are defined by the contents of the J table word (JTW). This word is located at the address defined by the BAR register. The format of the JTW is shown on Figure 11a. The size (JTSZ) 1101 or the J table 1204 on Figure 12 defines the number of entries in the J table 1204 which may be up to 255 entries. The JTSZ 1101 is an 8 bit positive integer; an out of J table exception occurs if J is greater than JTSZ. The absolute address of the J table 1204 is obtained by multiplying the J table pointer 1102 by 16. The J table 1204 contains J table entries whose format is shown in greater detail on Figure 11b. Each J table entry defines the absolute address of a P

table 1205 which is obtained by multiplying the P table pointer 1104 by 16. The size (PTSZ) 1103 of a P table defines the number of entries in the P table. The PTSZ is an 8 bit positive integer which may typically vary from 0 to 255 to indicate the number of entries in the P table. An out of P table exception occurs if P is greater than PTSZ. Each entry of the P table 1205 defines the absolute address of a process control block (PCB) 1206 by multiplying the process control block pointer 1107 by 16. A presence indicator P 1105 indicates the absence of a PCB 1206 when set to binary 0 and indicates the presence of a PCB when set to binary 1. (When the presence indicator P 1105 is found to be 0 a vacant P table entry exception occurs). Bits 1 through 7 of the P table indicator (Figure 11c) must be 0 (MBZ) 1106, otherwise an illegal P table entry exception occurs.

At address BAR plus 4 of the system base 600 there is the format byte of a G table word (GTW) shown in greater detail on Figure 11d. The size and the address of a G segment-table 1212 on Figure 1200 are defined by the contents of the G table word (GTW). The size (GTSZ) 1108 of the G table 1212 defines the number of entries in the G table which may typically be up to 255 entries. GTSZ is an 8 bit positive integer; an out of G table exception occurs if the G number is greater than the GTSZ. The absolute address of the G table 1212 is obtained by multiplying the G table pointer 1109 by 16. The format of the G segment table entry has a two word size (8 bytes) and is called a G segment descriptor. The format of the G segment descriptor is shown in detail on Figures 11e and 11f. All G segment descriptors are direct and therefore the indirect bit I, 1111 must be 0 otherwise an illegal segment descriptor exception occurs. The presence indicator P 1110 is a one bit field which when set to binary 1 indicates that a segment is defined in main storage for the segment number to which that descriptor corresponds;

whereas if it cleared to 0 no segment is defined and a reference to the segment descriptor causes a missing segment exception. The available bit A 1112 is a one bit field which indicates whether or not the segment is available; it is only checked if this segment is defined (i.e. P equals binary 1), otherwise it is ignored. The used flag field U 1113 indicates whether or not the segment has been accessed. If the U bit is set to binary 0 the segment has not been accessed; whereas if the U field is set to binary 1 the segment has been accessed. The written flag field W 1114 indicates whether the segment has been written. If W is set to binary 0 the segment has not been written; whereas if W is set to binary 1 the segment has been written. The gating indicator GS 1115 of a G segment descriptor must be set to binary 01, otherwise an illegal segment descriptor exception occurs. The reason for this is that a G segment always contains semaphores (although the reverse is not true i.e. all semaphores are not required to be in a G segment) and instructions on semaphores require the GS code to be binary 01. The absolute address of the base of a segment 1214 is defined in the G segment descriptor of Figure 11e by the 24 bit base field 1116; the content of this field is multiplied by 16 to obtain the absolute address. The second word of the G segment descriptor of Figure 11f occupies bit position 32 through 63 in the G table 1212. The RSU field 1117, bits 32 through 39 is reserved for software use and is generally ignored when used as a G segment descriptor as it is in this case. The MBZ field 1118 must be 0 otherwise an illegal segment exception occurs. Since the MBZ field 1118 occupies bits 40 through 51 it sets the SIZE field 1119 which is the field for a small segment SIZN; hence all G segments must be of the small segment type. The segment SIZN 1119 is a 12 bit positive integer defining the number of bytes in the segment and the segment size is interpreted as a multiple of 16. Therefore the segment size for

a G segment 1214 cannot exceed 2^{16} bytes (small segments).

Referring once again to the system base 600 of Figure 6 there are 9 system exception cell words located between BAR plus 8 and BAR plus 44. The format of the system exception cell words EXC is shown on Figure 11g. Since semaphores are utilized for transmitting messages to dedicated processes when a system exception occurs the pointers to these semaphores are found in 9 locations of memory each location called a system exception cell—one per class of system exception. The MBZ field 1120 must be set to binary 0 otherwise a system check occurs. Each exception cell (EXC) contains the system name G, D 1121 and 1122 respectively.

The channel exception cell located in BAR plus 44 of the system base 600 has a format which is similar to the system exception cell previously discussed and contains the system name GD of a semaphore which is used for transmitting messages to dedicated processes when a channel exception occurs.

An internal processor queue word IPQW is located beginning at BAR plus 48 and details of its format are shown on Figure 11h. The IPQW word points to the head of a queue of processes ready (Q/PR/RDY) as shown on Figure 9 by reference numerals 905 and 805. The queue of processes ready (Q/PR/RDY) links all processes which are in the ready state. It is referenced by the HEAD OF Q/PR/RDY-field 1124 (Figure 11h) of the IPQW word by pointing to the top of the ready process queue. The HEAD of Q/PR/RDY-field 1124 contains a 16 bit positive integer which is the displacement from the base of the G segment number 0, referred to as the GO segment, to the first byte of Q/PR/RDY. If this Q/PR/RDY bit field is 0, the ready queue is considered to be empty. The MBZ field 1123 must be 0 otherwise a system check occurs.

At BAR plus 52 of the system base 600 there is shown the storage for the initial and current retry counts whose format is shown in detail on Figure 11i. The NFS field 1125 is a nonfunctional storage field and is not utilized by the system base. The initial retry count field 1126 and the current retry count field 1127 are used to control the number of times automatic instruction retry is executed before a machine error is made to produce a machine failure exception condition. They are loaded with the same number by a Reset Retry Count (not shown herein).

The running process word (RPW), shown in Figure 11j, is stored in BAR plus 56 of the system base 600 and is used to store the name of the running process with its priority in case of a monoprocessor architecture. The NFS fields 1128 and 1131 respectively are nonfunctional storage fields and may be utilized for any purpose by any facility but is generally not utilized by the system base. The priority level of a running process is stored in the PRI field 1129. An asynchronous trap bit is stored in AB field 1130; whereas an asynchronous trap ring is stored in ARN field 1132. The logical name J, P of the running process in case of a monoprocessor architecture is stored in the J, P field 1133.

An Absolutization Table Pointer word shown on Figure 11k is located at BAR plus 60 in the system base 600 and is utilized in initial system load to initialize the absolute addresses in the initial system load (ISL) program by adding the contents of BAR to all absolute addresses in the ISL program. The Absolutization Table Pointer 1135 defines the location of an Absolutization Table (not shown). The Absolutization Table Size is shown by the ATSZ field 1134.

The CPU serial number word shown on Figure 11l is a 4 byte word located at BAR plus 64 and contains the serial number of the CPU in the CPU serial number field 1136.

A main storage upper limit word shown on Figure 11m is located at BAR plus 68 and indicates the main storage upper limit 1139 by providing the absolute address of the last available word in main storage.

At BAR plus 72 there is located a word shown on Figure 11n for providing the initial system load ISL device channel number (CN) 1140 and the hardware device channel number (CN) 1141.

The type and subtype of a device used in the computer system is shown by a hardware device type word (Figure 11o) in fields 1143 and 1144 respectively; where the RSU field 1142 is reserved for software. This word is found in the system base at BAR plus 76. A similar word having a similar type format shown on Figure 11p contains the type and subtype of the device used in the initial system load. This word is located at BAR plus 80.

When the restart button of a computer is pressed, a simulated V-operation is performed on a semaphore and the Ready state is entered. A pointer to this semaphore is found at BAR plus 84 of the system base 600 and is called a restart cell word, and has a format shown on Figure 11q. The format is similar to the system exception cell described supra and contains the system name G, D of a semaphore in the G field 1149 and D field 1150 respectively. The MBZ field 1148 must be 0.

When there is more than one processor to the computer system, a word is provided in the system base 600 at BAR plus 88 for multiprocess extension. Details of this word are shown on Figure 11r.

Examples of System Base and Process Control Block Use

Referring to Figure 12 there is shown one example, how the system base may be utilized in combination with the process control block in order to address and access a user

segment, a system segment, or a queue of processes read (Q/PR/RDY) segment. Main memory 1200 has a portion 1203 reserved for hardware use. A boundary address register BAR 1202 separates the system base 1215 from the portion of memory 1203 reserved for hardware. The boundary address register BAR 1202 is utilized to address items in the system base 1215 by adding the contents of the boundary address register to the displacement in 4 byte units of the item desired in the system base. This address then points to the first byte of the item in the system base desired. In Figure 12 the BAR 1202 is pointing at the J table word (JTW). The J table word, as previously discussed, has a pointer which points to a J table 1204. By indexing to the J number shown on Figure 5, a J table entry 1216 is obtained. At the J table entry there is a P table pointer which points to the absolute address of the P table 1205. By indexing to the P number (see Figure 5) within P table 1205 the absolute address of the process control block 1206 is obtained. As previously shown in process control block PCB 1206 there are two address space words ASW 0 and ASW 1. The high order bits of the segment table number field STN in the base register 1201 is used to access one of these two address space words, in this instance ASW 1 which has a segment table word array STWA pointer that points to segment table word array STWA 1208. Together with the segment table number STN of the base register 1201 one of 8 segment table words is accessed in STWA 1208, which points to one of 8 segment tables 1210. The segment table entry STE from base register 1201 is then utilized to make one of 256 entries in segment table 1210 where a segment descriptor is located. The segment descriptor is then utilized to access a user segment 1211. (For greater detail see Application No. _____ filed on _____ (entitled Segmented Address Development and assigned to the same assignee as the instant invention and is hereby incorporated by reference herein).

In order to access a system segment 1214 which is utilized to store semaphores a G table word GTW is utilized in the system base 1215. The address of the G table word is obtained by adding the displacement of the G table word in the system base to the boundary address register BAR 1202. (See Figure 6). The G table word GTW includes a G table pointer which points to a G table 1212. By utilizing a G number available to the system and indexing in the G table a G segment descriptor is accessed which is utilized to address a system segment 1214.

Similarly the system base 1215 is utilized to access the queue of process ready (Q/PR/RDY) 1213 by locating an internal processor queue word IPQW which points to the Q/PR/RDY segment 1213.

Control Unit

Referring to Figures 13a-13c details of the control unit are shown. The control unit, although shown separate from the central processing unit (CPU), is in actuality a part of the CPU and is comprised of a control store unit CSU 1301, a control store interface adapter CIA 1302 and appurtenant subunits, control store loader CSL 1303 and control and load unit CLU 1304.

The control store unit CSU 1301 receives micro-instructions from the control store loader CSL 1303 via the control and load unit CLU 1304 and the control store interface adapter CIA 1302. Under normal operating conditions, microprograms are loaded from an external source during system initialization and become a permanent control function of the machine. However the control store unit CSU 1301 has the ability to be reloaded and initialized in a manner that provides for a variety of central processing unit CPU 1306 operational modes. The

following modes of operation of the CPU are available under control of the CSU 1301; (a) native mode; (b) emulation mode; (c) concurrent native and emulation modes; (d) diagnostic mode. This capability is possible because the micro-instructions resident in the CSU are the source of micro-operations used to control the operation of all other CPU functional units such as the emulation unit 1316, the arithmetic logic unit ALU 1317, the instruction fetch unit IFU 1318, the address control unit ACU 1319 and the data management unit DMU 1321. Also shown within the central processing unit CPU 1306 are previously described general registers 1307, base registers 1308, scientific registers 1309, T-registers 1310, status registers 1311, instruction counter IC 1312, and hardware control mask register 1313.

Typically the control store unit CSU 1301 is a 9K bipolar integrated circuit programmable read-only memory (PROM) mixed with read/write random access store (RAM). It has a typical 150 nanosecond read cycle and a 450 nanosecond write cycle. Each location of control store stores one 84-bit micro-instruction word (to be later more fully described), and each micro-instruction word controls one CPU cycle. As each location of the control store of the control store unit CSU 1301 is read, its contents are decoded by micro-operation decoders which provide micro-operation control signals each of which causes a specific operation within the CPU to take place (to be later described in detail).

By grouping locations within each micro-instruction word (to be later described in detail) control store sequences are obtained that can perform a specific CPU operation or instruction. As each instruction is initiated by the CPU, certain bits within the op-code are used to determine the control store starting sequence. Testing of certain flops (not

shown) which are set or reset by instruction decode functions allows the control store memory to branch to a more specific sequence when necessary.

The control store interface adapter CIA 1302 communicates with the control store unit 1301, the data management unit DMU 1321, the address control unit ACU 1319, and the arithmetic logic unit ALU 1317 for directing the operation of the control store memory 1333 of Figure 13b. The CIA 1302 includes logic for control store address modification, testing, error checking, and hardware address generation. Hardware address generation is utilized generally for developing the starting address of error sequences or for the initialization sequence.

The data management unit DMU 1321 provides the interface between the CPU 1306 and the main memory and/or buffer store memory shown on Figure 1. It is the responsibility of the data management unit to recognize which unit contains the information required by other units and strobe the information into the CPU registers at the proper time. The data management unit DMU also performs the masking during partial write operations.

The instruction fetch unit IFU 1318 interfaces with the DMU 1321, the ACU 1319, the ALU 1317, and the CSU 1301, and is responsible for keeping the CPU supplied with instructions. The instruction fetch unit has the next instruction available in its registers before the completion of the present instruction. To provide this capability, the instruction fetch unit IFU 1318 contains a 12-byte instruction register (not shown) that normally contains more than one instruction. In addition, the IFU, under control of the CSU, requests information (instructions) from main memory before the instruction is actually needed, thus keeping its 12-byte instruction register constantly updated. Instructions are thus prefetched by

means of normally unused memory cycles. The instruction fetch unit also decodes each instruction and informs the other units of the instruction's length and format.

The address control unit ACU 1319 communicates with the IFU, ALU, DMU, and the CSU via the CIA. The ACU 1319 is responsible for all address development in the CPU. All operations of the ACU, including transfers to, from, and within the unit, are directed by CSU micro-operation and logic in the unit. The normal cycling of the ACU depends on the types of addresses in the instruction rather than on the type of the instruction. Depending on the address types the ACU may perform different operations for each address in an instruction. The ACU also contains an associative memory 1319a that typically stores the base address of the 8 most recently used memory segments, along with their segment numbers. Each time a memory request is made, the segment number is checked against the associative memory contents to determine if the base address of the segment has already been developed and stored. If the base address is contained in the associative memory 1319a, this address is used in the absolute address development, and a considerable amount of time is saved. If the base address is not contained in the associative memory 1319a it is developed by accessing the main memory tables. However, after the base address of the segment is developed, it is stored in the associative memory, along with the segment number, for future reference.

Interfacing with the ACU, IFU, DMU, and the CSU is the arithmetic and logic unit ALU 1317. Its primary function is to perform the arithmetic operations and data manipulations required of the CPU. The operations of the arithmetic logic unit are completely dependent on micro-operation control signals from the control store unit CSU 1301.

Associated with the ALU 1317 and the CSU 1301 is the scratch pad memory unit LSU 1315, (sometimes referred to also as the local store unit). It is typically comprised of 256-location (32 bits per location) solid state memory and selection and read/write logic for that memory. The scratch pad memory 1315 is used to store CPU control information and maintainability information. In addition, the scratch pad memory 1315 contains working locations which are primarily used for temporary storage of operands and partial results during data manipulation. Also associated with the ALU 1317 is an auxiliary memory 1317a comprised typically of 64 flip-flops for storing miscellaneous states of the computer system.

The CPU also has a clocking unit 1320 and is essentially 2 clocking systems in 1: the first clocking system generates the timing for the control interface adapter CIA 1302 and the second clocking system generates the timing pulses for the operations of the functional unit within the central processing unit.

Referring now to Figure 13c there is shown the format of the control store word 1325. The control store word is typically 84 bits wide and is divided into 6 main fields:

a. sequence type field 1326 (3 bits);
b. branching and/or micro-operations 1327 (23 bits);
c. constant generation and designation 1328 (14 bits);
d. data to bus 1329 (8 bits);
e micro-operations 1330 (32 bits); and
f. checking 1331 (4 bits).

The 3-bit F field of the control store word 1325 is used as a sequence control field. There are typically 7 different sequence types and 1 reserved type for the instant computer system. Referring to block 1335 of Figure 13b, when E field equals binary 0, 1, or 2, the branching field A, B, C, D, and I, of micro-instruction 1325 is utilized to generate the

next address. The first 6 bits of KS register 1337 is utilized together with the B field, a C test results, the D test results and the L field to provide the next address of the next micro-instruction which is then placed in address register KS 1337. When the E field is set to binary 4 (see block 1335) the next address selected is taken from interrupt return register KA 1339. The address stored in the KA register is the one generated by the next address generation logic when the hardware interrupt occurs. When the E field is set to binary 5 a branch is used to initiate a subreturn from a micro-program subroutine. When used, the contents of the return register KR 1346 are used as the next control store address. The return register 1346 is loaded by issuing a control store command which will load present control store address in KS register 1337 plus 1, from incrementor 1338, into the KR register 1346. A one-level-nesting subroutine ability is provided via the KT return branch register 1347. Every time the KR register 1346 is loaded the old contents of the KR register is transferred to the KT register 1347 every time the micro-program return is called; the contents of the KT register will transfer to the KR register. Third level nesting subroutine ability is provided by the KU register 1340; and fourth level nesting subroutine ability is provided by the KV return branch register 1349. When the E field of the control store word is set to binary 6 the next control store word addressed is equal to the present address in KS register 1337 plus 1 in incrementor 1338. When the E field is set to binary 7 the CSU 1301 enters the diagnostic mode and the next address will be the present address plus 1.

In addition to the sequencing control of branching to the next control store address described above and shown in block 1335, there is hardware generated sequence control shown in block 1336 of Figure 13b. (Note: Blocks 1335

and 1336 are in actuality hardware registers drawn so as to depict the different forms that the microinstruction words may take.) The hardware generated branches are overriding conditions (such as errors, initialize, control store scan, etc.) which surpass the E field and force a fixed address into the control store address register KS 1337. The branch is made by forcing an interrupt line high (not shown) for one clock period and storing the address which would have been generated under the control of the E field into the KA interrupt return register 1339. A hardware generated address will be placed into the control store address register. Certain hardware/firmware generated interrupts take priority when the interrupt-block flip-flop (not shown) which prevents additional interrupts in their class from being executed until the interrupting condition has been satisfied. A firmware micro-operation exists for controlling the resetting of the interrupt-block flip-flop for those sequences which are under firmware control. Those sequences under hardware control automatically generate a reset of the block-flop at the end of the sequence. The following conditions, listed by priority, exists in this category; (a) control store load; (b) control store scan; (c) hardware error; (d) software error. The remaining hardware conditions do not set the interrupt block-flop but do cause an immediate action to occur when generated. The following conditions listed by priority, exist in this category;

(a) initialize;
(b) soft-clear;
(c) enter maintenance panel;
(d) enter maintenance panel;
(e) hardware exit.

An initialize signal causes the CSU 1301 to branch to address binary 0, clear hardware resettable errors and execute a control store load operation followed by a control

store scan sequence under hardware control. It will also perform system initialize. A soft-clear signal causes the CSU 1301 to branch to address binary 0, clear hardware resettable errors and reset the interrupt block-flop. An enter maintenance panel signal causes the CSU to branch to the address preset in the CSU address switches on the maintenance panel (not shown).

An enter maintenance channel signal causes the CSU to branch to an address generated via the maintenance channel (not shown). The address loaded is from maintenance bus QMB 1344, which is part of the maintenance channel, and is right-justified. A hardware exit signal causes the CSU to branch to binary address 2. This sequence is used as a maintenance facility. At the end of the sequence a return is intiated by issuing an E field branch with the E field set to binary 4.

A control store load signal causes the CSU to branch to address binary 0. It also turns off the CSU read-cycle flop (not shown), the system clock 1320, and places the CSU in the load state. In the load state the CSU can be loaded from the control store loader CSL 1303, the IOC 1305, the main memory 102, or the maintenance panel 1355. When loaded from the CSL an automatic scan is generated at the end of the load. When loaded from any other media a scan may be issued by either generating a micro-operation signal or setting the scan switch on the maintenance panel. A control store scan signal causes the CSU to branch to an address binary 0. A control store scan is under hardware control for the duration of the sequence. During the scan the system clock 1320 is off and therefore no commands or tests are executed. At the end of the scan sequence the hardware transfers the contents of the interrupt return register KA to the address register KS, the system clock is turned on and control is returned to the firmware.

A hardware error signal causes the CSU to branch to address binary 4. In the normal processing mode a hardware error detected in any CPU functional unit will activate a hardware error line (not shown). The control store sequence generated will test the system conditions to determine the action to be taken. In the diagnostic mode, error conditions which are hardware detectable are visible to microdiagnostics. The microdiagnostics control the action to be taken. A software error signal on the other hand causes the control store to branch to address binnary 1. This address is the start of the software error reporting sequence which is under micro-program control.

Referring once again to Figure 13c the E field 1326 is a 3 bit field for the branch code as previously described. The branching and/or micro-operation field 1327 is comprised of the A, B, C, D, and L fields (also shown on block 1335 of Figure 13b) wherein the A field is the upper 6 bits of the next address, the B field is the middle 4 bits of next address of the mask field 64 tests, the D field is another 6 bit test field for 1 of 64 tests, and the L field is the least significant bit. The K field 1328 is a 14 bit field of which 6 bits are for the constant field, 4 bits are for a constant or steering field, and 4 bits are a steering field for a constant. The data to bus field 1329 is comprised of the QA field having 4 bits for controlling information to the QA portion of the QMB bus 1344 and the QB field has 4 bits for controlling information to the QB portion of the QMB bus 1344. The F field 1330 is a 32 bit field which is coded to generate micro-operation subcommands. The P field 1331 is comprised of 4 bits reserved for checking.

In operation the micro-instruction words are stored in the control store array 1333. During a cycle of operation, the control store array is addressed by the contents of the KS address register 1337. This causes the contents of the

location specified by the address to be read into the group of read latches 1357. Portions of the word contents of the read latches are distributed or transferred to storage registers within each of the functional units in the CPU. Each functional unit includes decoding logic circuits for generating the requisite subcommands specified by the control store word under control of the system clock source. In general decoding is performed within each functional unit in the CPU rather than being performed centrally in order to minimize the decoding time and to reduce the number of cables which would be normally required for transmitting command signals if decoding were performed centrally. Additionally, the decoding is done within each unit to avoid timing problems arising from differences in cable delays. Further, by decoding subcommands with each unit, those signals which are representative of certain conditions existing within the functional unit are required for the generation of certain subcommand signals do not have to be returned to the CIA unit 1302. A typical decoder unit 1359 is shown in Figure 13b as receiving various fields from micro-instruction words and generating micro-operation signals a, b, c, d, . . . q, r. A typical micro-operation decoder 1359 receives commands from a micro-instruction word. The field from the micro-instruction word is decoded and sets one of a plurality of lines s, t, u. . . . y, z high. A matrix is formed by having predetermined control line impedance coupled to the s-z lines at points α, β, $\gamma \ldots \psi$, ω. Typically then when the field from a micro-instruction is decoded one of the lines s-z goes high. Since the black dots shown in the matrix by Greek letters α through ω represent impedance coupling between the two sets of lines, any electrical signal propagating along any horizontal wire will be coupled through to propagate along the vertical wire a-r where an impedance coupling (black dot) is indicated.

Each vertical line a-r then may be coupled as one input to one each of AND gates 1360-1365. Other input signals may also be coupled to AND gates 1360-1365. Other input signals may also be coupled to AND gates 1360-1365 including a timing signal t_s from the central timing unit. Accordingly as each timing signal t_s goes high those gates which have all the other input signals high will be enabled and will provide micro-instruction signals to predetermined functional units in the CPU. For example, if a command 1341 from read latch 1357 is decoded and a horizontal line is high the a, b, c, and q vertical control lines will be high and AND gates 1360, 1361, 1362, and 1364, will be enabled as the t_s timing signal is applied to these gates sequentially. Hence the combination in which the vertical control lines are coupled to the horizontal control line at different points represented by the Greek letters α through ω represent a permanent switching matrix for supplying micro-operation signals to the central processing unit CPU for controlling the functional units within the central processing unit by micro-instructions furnished from the control store array 1333. Thus permanent firmware having an alterability feature can be built into the machine of this invention by merely specifying the sequence of micro-operations that is required as a capability of the computer system.

Under normal conditions data is written into the control store array 1333 via the CPU write-data register also known as the local register YO 1343. A control flop (not shown) defines whether the upper half or lower half of the storage array is to be written. Data from the control and load unit CLU 1304 arrives at the CIA/CSU via the maintenance bus QMB 1344 and is buffered by the storage local register YO 1343 before being written into the control store array 1333. The storage local register 1343 is time

shared as both a read and a write local register. The multiplexor KQM 1345 may be controlled by either the maintenance panel 1355 or by micro-diagnostics and provides a read out path from the registers connected to it. A compare register KP 1350 is provided for non-functional use and is used mainly for maintenance purposes and is utilized along with compare logic 1352 and decode logic 1351.

DISPATCHER FIRMWARE FOR CONTROL OF PROCESSES

The dispatcher is a firmware/hardware unit whose main purpose is to manage the various queues of processes, and to switch between processes and includes the updating of the queues of processes, the process control blocks PCB's, the running process word in the system base and the register of a new process. It also delivers messages to a process which is actually waiting for it on a semaphore (after a V-operation, a simulated V-operation for the IOC or for the exception handler). It also enqueues a message on a semaphore after a P-operation which freed a message link when a process is waiting on the free-link semaphore to deliver its message.

The dispatcher unit further calls on the native mode instructions firmware after the "rolling-in" of a process executing in native mode, or after a "contest" if the current process remains running and is executing in native mode. It also calls on decor extension firmware for:

(a) a temporary call during the rolling-out of a process executing in that decor extension;

(b) a temporary call during the rolling-in of a process executing in that decor extension;

(c) a definitive call at the end of the rolling-in of a process executing in that decor extension; and,

(d) a definitive call after a contest if the current process remains running and is executing in that decor extension.

Moreover the dispatcher places the system in the idle loop when there is no running process.

There are several ways of entering or leaving the dispatcher as follows:

(a) The initialization procedure (SIP) provides entry as the last step, (See Patent Application No. _____ entitled System Initialization Procedure, assigned to the same assignee named herein.)

(2) The Start and Suspend Instruction provides for the entry into the dispatcher. The Start instruction starts a process and the Suspend instruction terminates a process. (See Patent Application No. _____ entitled Start and Suspend Instruction.)

(3) The P and V operations provide for entry into the dispatcher. The P op picks up a message from a semaphore and if no message then the process goes into wait state. (See Patent Application No. _____ entitled "P and V Operations of Semaphores".)

Briefly then, the dispatcher is the main mechanism which manages the processes and accordingly the process control blocks PCB's by deciding which process is to be run and then taking the appropriate action, such as rolling-out the currently running process (i.e. write into the PCB all information relative to the currently running process contained in hardware registers, scratch pad memory, etc.), and rolling-in the new process (i.e. write from the PCB to various hardware registers, scratch pad memory, etc., all information necessary to run the new process.)

The representation of the functions performed by the dispatcher are shown in flow-charts Figs. 14a-14i. For

example, block 1402 of Fig. 14a is a representation of the function performed by the dispatcher wherein a micro-program word is delivered by the control store unit, and upon decoding by decoder 1359 controls the applicable portions of the CPU via an appropriate series of micro-operation signals 1360, 1361, etc. to retrieve the IPQW from the system base in storage subsystem 102, and transfer it to scratch pad memory 1315. Simultaneously the dispatcher fetches 1404 the GO segment descriptor (see Figure 12) from the G table of segment descriptors pointed to by the G table word GTW in the system base. Bits 16-31 of the IPQW word contain a 16-bit positive integer which is the displacement from the base of the G segment number 0, referred to as the GO segment, to the head (first byte) of the Q/PR/RDY, queue of processes ready. If bits 16-31 of the IPQW word is 0, 1403 the ready queue is considered to be empty. If the ready queue is empty it indicates that there is no process currently waiting in the Q/PR/RDY and the ready-queue is empty. The next question to be decided in decision block 1405 is whether or not a process is currently running in the machine by determining whether or not the Vacant Indicator is set. (The Vacant Indicator is a flip-flop located in auxiliary memory 1317a; it is set when there is no current process CJP running in the processor). If the Vacant Indicator is set (i.e. there is no process currently running) and since it was previously determined that there is no process in the ready-queue waiting to use the processor then the machine goes in the idle state 1406. However if there is a process currently running in the machine, but there is no one waiting to use the machine, the current process accesses its next instruction 1407.

Returning now to decision block 1403, of the flow chart of Fig. 14a if there is a positive integer in the pointer area of the IPQW (ie. bits 16-31) the head of the ready queue

pointed to by the IPQW word in the GO segment is fetched to the scratch pad memory. (Note: To avoid repetition and in the interest of clarity the intermediate functions of the dispatcher in conjunction with the control unit and CPU will be omitted; however it is to be understood that such intermediate functions as previously described by example are present typically.) Up to this point it has been determined that there is some process waiting in the ready-queue. Before further action can be taken it is necessary to determine if there is a process currently running in the central processor. This is determined at decision block 1410 of the flow chart representation and if there is no process currently running in the central processor (i.e. no CJP) the head of the ready queue is run 1412. However if there is a process running in the central processor the dispatcher must determine who has priority — the currently running process or the head of the ready queue? Accordingly therefore the priority byte of the current process (CJP) which is located in the running process word of the system base 600 or in process main word 0 PMW 0 of PCB 400, is fetched 1413. A decision is then made 1414 as to whether or not the current running process CJP is of lower priority than the new process NJP waiting at the head of the ready queue? (See decision block 1414.) If CJP is not of lower priority than NJP, CJP remains in control of the central processor and the contest indicator is reset 1415. (The contest indicator is always set to 0 except when one or more new processes have been placed on the ready-queue since the beginning of the last instruction executed on behalf of CJP thereby creating the possibility of a conflict; under these conditions the contest indicator is set to binary 1). Before the current process CJP is permitted to continue however, and execute further instructions a determination is made whether or not the CJP is running in decor extension

mode 1415. If CJP is running in decor extension mode then the next instruction is executed in emulation mode (i.e. decor extension) and if it is not running in decor extension mode then the next instruction is executed in native mode. Returning once gain to decision block 1414, if the NJP at the head of the ready queue has a higher priority than the CJP (i.e. its priority number is lower than CJP's priority number, the current running process CJP is "rolled-out" of the machine and the new process NJP is "rolled-in" to the machine. Accordingly a firmware priority subroutine PRIQ 1418 directs the enqueueing of the current process CJP into the ready-queue by LIFO priority and by priority number by first "rolling-out" the CJP under the direction of the firmware subroutine RLLO 1419. The RLLO subroutine directs the writing of the information of the CJP that is stored in the general registers, base registers, scientific registers, T-register, status registers, and instruction counter, back into the appropriate storage areas of the process control block PCB in main memory, and it directs the updating of the RUA. Moreover the DEXT number of process main word 0 (PMW0) in PCB 400 is updated 1420.

The new process NJP is now ready to be "rolled-in". The boundary address register BAR is fetched 1422 and the running process word RPW is fetched from address BAR plus 56 of the system base. See block 1423. The name of the new process NJP is next written into the running process word RPW and since the name of the new process NJP was written in the process link PL of the Q/PR/RDY, the name in the process link PL is therefore now placed in the RPW, block 1424. Therefore NJP from the ready-queue now becomes CJP and is entitled to control the central processor and accordingly is no longer waiting in the Q/PR/RDY and must be dequeued by taking its name out of the process link

PL of the Q/PR/RDY, block 1425. When this is done the queue of processes ready Q/PR/RDY is updated by the firmware subroutine UQLK, 1425a. Accordingly the JP number of the process which was just taken out of the machine is placed in a process link in the Q/PR/RDY, since it now no longer has control of the machine and must wait for it, 1426. At this point the transformation of giving this control of the central processor to the new process and placing the old process in a ready queue is accomplished and because there is a process (the new CJP) in control of the central processor, the Vacant Indicator is set to 0, 1427; if on the other hand there were no CJP in control of the central processor the Vacant Indicator would be set to 1. At this point the assignment of the processor is complete and a new process has acquired the central processor whereas the old process was placed in a ready queue; however, the new process is not ready to run yet for the hardware of the central processing unit, 1306 of Figure 13a, such as the general registers 1307, the base registers 1308, scientific registers 1309, the T-register 1310, the status registers 1311, and the instruction counter 1312, must be supplied with control information from the process control block of the new process.

Accordingly the firmware subroutine 1430 first fetches PMW 3 from the PCB (Figure 4) to scratch pad memory 1315, and then fetches PMW 0. The MBZ field of PMW 0 is checked 1433 and if it is not binary 0 an illegal PCB exception results. However if the MBZ field of PMW 0 is zero then PMW 1 is fetched 1434. Again the MBZ field of PMW 1 is tested to determine whether or not it is a binary 0. If it is not binary 0 there is an illegal PCB exception; whereas if it is equal to 0 the dispatcher proceeds to C.

Accordingly address space word 0 ASW 0 is fetched from the appropriate space in the PCB and the segment table

word size STWSZ is tested 1437 to determine whether or not it is less than 7. If it is greater than 7 an illegal PCB results if it is less than or equal to 7, then ASW 1 is fetched from the PCB, block 1438, and its STWSZ field is tested 1439 to determine whether or not it is less than or equal to 8. If its field is greater than 8 then an illegal PCB results; however, if its STWSZ field is equal or less than 8 the exception word EXW is fetched 1440, and its MBZ field is tested to determine whether or not it is equal to 0. If its MBZ field is not equal to 0 an illegal PCB results whereas if it is equal to 0 the stack word SKW is fetched 1442, and its MBZ field is tested 1443 to determine whether or not it is equal to 0. If the MBZ field is not equal to 0 then an illegal PCB results; whereas if it is equal to 0 the instruction counter word ICW is fetched from the PCB and placed in the instruction counter IC and its TAG field is tested to determine whether or not it is equal to 0, 1445. If the TAG field is not equal to 0 then an illegal PCB results. However if the TAG field is equal to 0 the MBZ word is fetched 1446 and its MBZ field (bits 0-31) is tested to determine whether or not it is equal to 0, 1447. If it is not equal to 0 then an illegal PCB results; whereas if it is equal to 0 stack base words 0, 1 and 2 SBW 0, 1 and 2 are fetched 1448. The contents of the 8 base registers in the base register saving area of the PCB is then fetched 1449 and stored in the machine's base registers 1308. Then the contents of the 16 general registers from the general register saving area of the PCB are fetched 1450 and stored in the machine's general registers 1307. Before fetching the contents of the scientific registers however a check is made of the capability byte of process main word 0 (PMW 0) to determine whether or not scientific mode is being utilized 1451. If scientific mode is being utilized then the contents of the scientific registers from the scientific register saving area of the PCB are fetched and stored 1452. The firmware then

proceeds to make a check of the capability byte of PMW 0 to determine whether or not the accounting mode is being utilized 1453. If the accounting mode is being utilized (i.e. accounting bit of capability byte set to binary 1) the accounting words exists in the PCB and the ready time account word RTA is updated. Then the firmware proceeds to determine whether or not the DEXT number is set to 0, 1454. If it is not set to 0 it indicates that the machine may be in emulation mode (i.e. the decor extension capability is being utilized) and accordingly the DEXT number of PMW 0 is checked 1455 to determine whether or not it is greater or less than the DETSZ field of process main word 3 and if it is greater than the DETSZ field an illegal PCB exception 1456 results since the DEXT number is less than the DETSZ field but not equal to zero the machine is performing in legal emulation mode and proceeds to F. Returning to decision block 1454 if the DEXT field is binary 0 then native mode is being performed and the machine fetches STW's, 1457. The residual time out word RTO word of the PCB is fetched 1458 and the process timer is loaded, with the time limit that the CJP may spend in the running state.

Up to this point either (a) a new process NJP has been "rolled-in" to take control of the CPU when there was an old process CJP in the machine and the new process NJP was at a higher priority than the old process CJP or (b) there was no CJP in control of the CPU and the head of the ready queue was rung. Briefly under condition (a) the CJP was taken out of the RPW and placed in a process link PL in the Q/PR/RDY, and the NJP in a process link PL in Q/PR/RDY was placed in the RPW thus effectively switching the positions of the 2 processes giving control to the NJP which now becomes the CJP and taking control away from the old CJP. Then the PCB of the NJP was accessed and the information required to run NJP (now the

CJP) was placed in scratch pad memory or the array of registers in the ACU.

If there was no CJP in control of the CPU (condition b), then the head of the ready queue was run—i.e. the NJP became the CJP because of the dispatcher taking the NJP from the process link PL from the head of the ready queue and placing it in the RPW word. By doing this a process link PL was left empty in the Q/PR/RDY and it is required to take it out.

Accordingly beginning now at decision block 1461 the firmware determines whether or not there was a CJP in control of the CPU and if there was a free process link (FPLS) it was accessed and enqueued and the CJP was written therein. However if there was no CJP in control of the CPU state byte of PMW 0 of the NJP is updated 1460 and again there is a determination whether or not there was a CJP in the machine 1463. If there was no CJP in control of the processor the process link of the NJP (which was in the Q/PR/RDY and is now in control of the machine) is taken out of the Q/PR/RDY 1466 (i.e. dequeued from the Q/PR/RDY) and becomes a free link semaphore FLSP and is now enqueued in the free process link queue (805 on Figure 9) and becomes a part of the free process link queue 1466a. The contents of the boundary address register BAR are fetched 1464 and the running process word RPW of the NJP (now the CJP) located at BAR plus 56 of the system base is updated by placing the NJP identification in the RPW, block 1465. The Vacant Indicator is set to 0 in the case where there had been no CJP. Next the Contest Indicator (i.e. a flip-flop in auxiliary memory 1317a used to indicate a possible conflict in priority between the CJP and a process placed in the ready queue) is set to 0, 1467 and the Segment Associator (AS 132 on Figure 1) which is a typical content addressable memory is cleared, 1471; then the

process mode is entered 1470. (A process mode indicates that exceptions are handled by the process which is operating in the processor rather than by the operating system). The firmware then continues to CAB 1480 and the asynchronous trap bit AB is checked to determine whether or not it is set to binary 1, 1481. If the AB bit is set to binary 1 then a check is made 1482 to determine whether or not the process ring number PRN is greater than or equal to the asynchronous trap bit ARN. (The AB and the ARN are located in the priority byte of the PCB of each process and is meaningful when the process is in the running state; the AB and ARN are obtained from the RPW located at BAR plus 56 of the system base). The AB and ARN in BAR plus 56 of the RPW are reset since the next step 1484 is to proceed to an asynchronous trap routine which would take care of the conditions that cause the asynchronous trap bit or the asynchronous ring number to be set in the first place, and if these were not reset then at the next pass by the firmware it would give an indication that something was wrong where actually there was nothing wrong, and hence would always proceed to the asynchronous trap routine 1484 and would never execute. Going back now to decision blocks 1481 and 1482 if the AB bit is not set or AB bit is set and PRN is not greater than ARN, then the firmware would proceed to determine in what mode the processor would run—in the normal mode or in the emulation mode. Accordingly the DEXT number is checked to determine whether or not it is set to 0 and if it is set to 0 the normal mode of the machine is run 1487. However if the DEXT number is not set to 0 the emulation mode is run 1486.

The Switch System Base Instruction

Referring now to Figure 15a there is shown the structure of the switch system base instruction. The operation code OP

1501 is similar to the operation code of any instruction and indicates the operation to be performed. The complementary code C 1502 is used in conjunction with the op-code 1501 when the op-code shares more than one instruction 1501 and the complementary code 1502 determines the precise instruction to be performed. For example, the P and V operations have 12 instructions and they will use one op-code. The complementary code together with the op-code determine which of the 12 P and V instructions is to be performed. The address syllable AS 1503 of the switch base instruction is the address pointing to a new copy of the system base which is to be loaded into main memory and also scratchpad memory. Accordingly, this address points to an entirely new system base. Typically a 60 byte zone, starting at the address specified by the address field of the switch system base instruction, is transferred into the system base area which starts at address BAR. This transfer is also made into the scratchpad memory. This zone, therefore, is the new system base which replaces the previous system base. The running process, therefore, disappears because its PCB is not updated and no link manipulation is performed. The processor is then placed in the VACANT state, and the dispatcher entered.

It is the system software responsibility to prepare the new information for the system base and the new GO segment before executing the SWSB instruction. (See applications included by reference herein numbers 12-16). For that reason, the SWSB instruction must be entered with the hardware gate HG locked, otherwise it returns a condition code CC without taking further action. Upon completion of SWSB instruction the HG will be left unlocked. Also upon completion of the SWSB instruction the vacant switch is set to 1. This indicates no process is running and as a result the system will go to the head of the ready queue for the internal

process queue word IPQW and pick out the pointer to the queue of processes ready. It will then proceed to the head of the ready queue and start up a new process. The process which executed the SWSB instruction will terminate because a vacant condition has been placed in the vacant indicator, which means that there is no running process.

Referring now to Figures 15b and 15c the SWSB instruction in firmware will be described in greater detail. The SWSB instruction starts 1504 and a determination is made 1505 whether or not the process ring number PRN is 0. This is necessary so that the SWSB instruction executes with the highest privilege. If the PRN is not 0 then a privileged instruction exception 1506 occurs. If on the other hand PRN is equal to zero the hardware gate located in auxiliary memory 1317a of the ALU 1317, is checked 1507 to determine whether or not it is locked. It is important that the HG is locked during the execution of the SWSB instruction. When the HG is locked no I/O interrupts or exceptions are permitted. If such exception or an error condition occurs while the HG is locked the system will crash. The system crash is a more desirable alternative while the system base is being changed, then permitting the interrupt with the system base only half changed. Accordingly, if the HG is not locked the condition code CC in one of the status registers is set to 2, (block 1508) which in this case is the code indicating the hardware gate HG is not locked.

Accordingly, the SWSB instruction is not allowed to proceed 1509, until the hardware gate HG is locked. If on the other hand the hardware gate HG is locked a determination is made as to whether or not the new system base word (i.e. address syllable AS) is quad-word aligned. If the AS is not quad-word aligned then the condition code CC is set to 3 indicating this condition and the SWSB instruction terminates 1509. On the other hand if the AS is quad-word

aligned the condition code CC 1512 is set to 0 which indicates that the SWSB instruction was successful and can proceed to perform the move operations 1513 from the new system base image into the system base area located at BAR. Finally after the new system base image has been transferred into the BAR system base the hardware gate is unlocked 1514, the vacant indicator in the auxiliary memory 1317a is set to 1, (block 1515) indicating no running process, and finally the instruction is complete 1516.

Figure 15c is a more detailed flow diagram of block 1513 which is the step that transfers the new system base image to the BAR system base area, and includes loading of the scratchpad memory S.P.M. with information from new system base image.

Referring to Figure 15c the start 1517 of SWSB instruction is actually at block 1513 of 15b. Typically 60 bytes of the new system base image located at address AS are transferred 1518 to the system base whose address is indicated by the boundary address register BAR. A new G table word GTW from the new system base image is loaded 1519 into scratchpad memory S.P.M. in auxiliary memory 1317a. Then a new J table word JTW is loaded 1520 in S.P.M. Finally a new internal processor queue word IPQW is loaded 1521 into S.P.M., and that portion of the instruction completes 1516 and continues to block 1514, where the operations are as previously described.

GLOSSARY OF TERMS

Absolute Address	- the physical address of a hardware byte in main memory store.
Address Development	- a hardware function which operates on a number of address elements to compute an absolute address which is used to refer to a location in main storage.
Addressing	- locating an object by any of a number of virtual, logical, and physical means.
Address Space	- the set of logical, segmented addresses corresponding to a process that the CPU is permitted to transform into absolute addresses during execution of that process.
Address Space Word	- one of two words in a process control block which point to the segment table word array. The segment table word array defines the segment tables associated with the process.
Address Syllable	- a logical address recognized by CPU hardware, normally an operand of an instruction.
Analyzer/Translator	- that part of the static linker that first passes the control language stream to the linker; then, if the control language is error-free, the analyzer/translator converts it to tables and other structures for the static linker to use.

Asynchronous Processing	- concurrent activation of more than one process.
Auxiliary Memory	- is comprised of 64 flip-flops for storing miscellaneous states of the system and is located in ALU.
Base Register	- the primary element in segment addressing, referenced by number in every address syllable.
Boundary Address Register	- a visible hardware register that defines the lowest memory address accessible to firmware/software.
Call	- see procedure call - Inward Call-Outward Call.
Central Process Unit	- a part of a computer that includes circuits controlling the interpretation and execution of instructions.
Channel	- a means of communicating between the processor subsystem and a peripheral subsystem. There are two types of channels, physical and logical. A physical channel is the hardware connection between the input/output control unit IOC and the peripheral control unit PCU. A logical channel is a software communications path between main memory and a single peripheral device.
Channel Command Entry	- an instruction in a channel program.
Channel Command Word	- an element of a channel command entry. Two channel command words constitute a channel command entry.

Channel Program - a sequence of instructions that cause a specific input/output i/o operation to be performed by a peripheral device.

Complementary Code - a code in some instructions which together with the operation code defines the function of the instruction.

Compilation Unit - the single object-code module that results from compiling or assembling one procedure in any high level language processor. The CU is not executable unit it is linked.

Concurrency - apparent simultaneity.

Condition Field - a 4-bit field in a branch instruction. Each bit of the field corresponds to a setting of the condition code in the status register bits 0 and 1.

Contest Indicator - located in auxiliary memory and indicates a new entry was placed in the ready queue.

Data Address Space - the set of logical record addresses accessible by a process through data management; consists of four elements; logical fields, logical records, data base pages, and files.

Data Descriptor - an information structure which is used in indirect addressing and as a description in data manipulation.

Decor - the functional characteristics or architecture of a computer system.

Decor Extension Mode - a mode of operation permitting the operating of the system in emulation mode.

Descriptor	- a 32- or 64-bit field used in developing the address of any data field. It is one of many pointers in the addressing scheme of the invention.
Descriptor Block	- a data block containing any number and variety of descriptor declarations except semaphore descriptors. No other data is allowed.
Device Adapter	- electronics which adapts a specific device to a peripheral controller.
Device Adapter Interface	- the interface between a device adapter and its associated peripheral controller(s).
Device Control	- a software facility which handles device specific characteristics.
Dispatcher	- the firmware responsible for dispatching processes in the CPU.
Dispatching	- a field within the instruction format or a data descriptor which provides relataive position information within a segment.
Displacement Addressing	- a form of direct addressing in which the operand defines a location relative either to the base of the immediate segment or the current ation counter.
Dynamic Addressing	- addressing which results in resolution, during execution, of references defined within the process group.
Editor	- that part of the static linker that format and outputs any or all of the information that results from linking a particular procedure.

Effective Address	- a logical, as opposed to physical, address consisting of a segment table number, a segment table entry, and a segment relative address. These elements show the way to a particular segment descriptor, which eventually points to the segment being referenced.
Emulation Mode	- operation of a computer so as to initiate another computer, and appear to operate in the other computer's environment.
Exception	- an exception occurs when the hardware detects the existence of some condition that has been specified as requiring special handling.
Event	- anything observed in a system by a process which may be of interest to some other process.
Firmware	- that part of a hardware unit implementation which utilizes a form of micro-programmed control.
G, D Name	- the name used by the hardware to refer to a semaphore. G, the G-segment name is the number of the entry in the G-table containing the segment descriptor; D is the relative address in the segment.
G-Segment	- a segment containing semaphores which may be addressed using the G, D name form and associated mapping facility (the G-table).

G-Table	- a table used for storage mapping (locating a G-segment) when the G, D name form is used.
Gating	- a facility for controlling entry to procedures in a (gated) segment through the use of procedure descriptors.
General Register	- a 32-bit register available to executing processes which generally holds binary or bit string data. Certain general registers can be used for indexing (GR8 through GR15).
Index	- a data structure which is maintained and operated upon by the system. The structure is visible to the user only in terms of a key which he presents for the purpose of making available or positioning to a specific record occurrence.
Indexing	- modification of an address by a signed arithmetic value by adding something to something.
Index Register	- a general register used for indexing (GR8 through GR15).
Indirection	- a method of addressing whereby an address is used to obtain another address, rather than the actual data.
Indirection to Base	- the form of indirection whereby a data descriptor containing a base register number and a displacement is found at the address referenced.
Indirection to Segment	- the form of indirection whereby a data descriptor containing a segment address is found at the address referenced.

Indirect Segment Descriptor	- a type of segment descriptor which contains information for addressing another segment descriptor, rather than a segment.
Instruction	- the unit of execution of a central processor visible to a programmer.
Instruction Counter	- a register associated with an executing process which contains the segmented address of the next instruction of the procedure being executed.
Interleave	- sequential accessing of memory modules to reduce memory access time.
Interrupt	- the discontinuance of a process in the system due to an event occurrence.
Interrupt Process	- a process which is activated in response to an interrupt.
Inward Call	- when one procedure execution in one ring number calls another procedure to execute in a lower ring number.
I/O Controller	- a unit which provides the basic control for a specific I/O subsystem.
I/O Processor	- potentially asynchronous system processes which deal primarily with the movement of data between peripheral storage or I/O devices and main storage.
J.P. Tables	- a collection of logical addresses for locating a process control block.
Job	- a unit of work for the system.
Job Control Language	- a language for describing a job and its component parts.

Job Step	- the major subdivision of a job leading to the execution of a program defined by a JCL statement.
Job Step Table	- a system table used to map the first part of a process name, i.e., to locate the process group table.
Linked Module	- the output of the static linker. It is a consolidated group of compilation units (CU) whose cross-references were resolved by the static linker.
Linking	- see static linking.
Logical Channel	- see channel.
Logical Processor	- the collection of hardware resources and control information necessary for the execution of a process.
Main Storage	- all addressable storage from which instructions can be executed or from which data can be loaded directly into registers.
Magnetic Tape Controller	- the elements of the peripheral subsystem related to magnetic tape devices.
Mass Storage Controller	- the elements of the peripheral subsystem related to mass storage devices.
Memory Management	- operating system facilities for the allocation, relocation, and disallocation of physical memory.
Message	- information transmitted from a source or to a destination—neither being a file.
Microinstruction	- same as microinstruction word, and control store word.

Micro-program	- a set of machine codes used to implement the control functions of a processor.
Multiplex	- to share a hard resource, such as memory, usually by time division.
Multi-programming	- the concurrent execution of two or more programs by a single computer.
Native Mode	- operation of a computer in its native environment.
Offset	- in address development, the number of bytes after the beginning of the segment at which an addressed part of the segment starts.
Operating System	- a system of basic software designed to support the efficient operation of user software.
Outward Call	- when one procedure executing in one ring number calls another procedure to execute in a higher ring number.
Peripheral Control Unit	- a stand-alone micro-programmed processor which executes channel programs to perform I/O operations.
Peripheral Subsystem Interface	- a standard interface used for transfer and control between free-standing peripheral control units and I/O control units.
Physical Channel	- see channel.
Physical Data Structure	- a complete definition of the organization of data as it is physically recorded on media.

Physical I/O - that part of the operating system which initiates and controls the transfer of data between memory and peripheral or terminal devices.

Physical Storage - hardware used for the storage of data. It is composed of various types of recording media and the recording/reading hardware.

P-operation - a machine instruction involving a semaphore which causes a process to enter the wait state or to receive a message.

Privileged Instruction - an instruction which must be executed in ring 0.

Procedure - a named software function or algorithm which is executable by a processor without concurrency: a Fortran subroutine, a Cobol program, a PL/1 internal procedure.

Procedure Call - a mechanism for creating a stack frame within a stack segment.

Procedure Descriptor - a word containing the location and entry point of a procedure.

Procedure Segment - the type of segment whose contents represents a procedure, procedures, or part of a procedure.

Process - the ordered execution of instructions without concurrency by a processor, either central or i/o.

Process Address Space - the memory addresses which may be referenced or to which control may be transferred during a particular process.

Process Control Block	- a data structure (hardware defined and recognizable) that contains information needed to specify the state of a process at any given time.
Process Group	- a related set of processes, usually those necessary for the performance of a single job-step.
Process Group Builder	- that part of the static linker whose actions yield the linked module.
Process Group Loader	- an operating system facility that makes final resolutions in a linked module to create an executable process group then it loads the process group into memory and starts its execution.
Process Group Table	- see P Table.
Process Link	- an entry in a ready process or waiting process queue.
Process State	- the dynamic condition of a CPU process e.g., running, ready, waiting, or suspended.
Process Switch	- the hardware function that disassociates one process from the central processor and associates another.
Process Synchronization	- the functions that act to coordinate the asynchronous activity among processes. These functions generally involve semaphores and the P and V operations.
Processor	- a unit capable of receiving data, manipulating it, supplying results, and controlling its sequence of operations in response to stored

programs: short for central processor, generic for central processor, peripheral processor, or hardware/ software processor.

Program - the specification of the procedural and associated information necessary to solve a problem, i.e. the ordered collection of instructions which is executed by the computer and operates on data to perform a user's job or some phase of that job.

Program Execution - the activity of a process in conformance to program specification.

Protection - the function, enforced by hardware and software, of preventing processes from interferring with each other or sharing each other's address space in an unauthorized manner.

P-Table - a hardware defined data structure containing entries, each pointing to the origin of a process control block. The P-table entries point to the total set of PCB's comprising a process group at any given time. Also called process group table.

Q/PR/RDY - queue of processes ready i.e. a ready-queue.

Queue - an ordered list of entities awaiting information, process or availability.

Ready State - a process state in which no processor is associated but all necessary resources, except a processor, are available for changing to the running state.

Relative Address	- the incremental/decremental location of some object relative to another.
Relocation	- movement of a segment from one location in main storage to another including the adjustment of all required references to its previous location.
Return	- the function and operations necessary to resume the action of a procedure at a point immediately following the point where it entered some other procedure.
Ring	- a protection attribute of a segment that restricts the read, write, and execute access of process to that segment. A level of privilege of a process for read, write or execute.
Rolling-in	- performing the operation necessary to give control of a processor to a new process.
Rolling-out	- performing the operations necessary to take away control of a processor by a process.
Running State	- the dynamic condition of a process (connected to a CPU) that is executing.
Scratch Pad Memory	- 256 locations in ALU for storing CPU control information-also known as a local store unit.
Scheduling	- determining the sequence of operations.
Scientific Registers	- a 64-bit register used in the manipulation of binary floating

point numbers. There are four scientific registers in the invention when the system has the scientific option.

Segment - a contiguous main memory space treated as a unit.

Segmentation - the use of a division of main memory into logical groupings called segments rather than into a single linear store.

Segment Base - the current origin of the segment. This is one of the fields in a segment descriptor word.

Segment Descriptor - entry in a segment table defining the characteristics of a segment or pointing to a segment descriptor.

Segmented Address - see effective address.

Segment Number - the identification of a particular segment consisting of a segment table number (selecting one of the segment tables of the process) and a segment table entry of that chosen table.

Segment Relative Address - during address development, the final value to be added to the segment base to obtain the absolute address.

Segment Table - a table containing segment descriptors used to establish the addressability of a process to its segments. The collection of segment tables of a process defines the address space for that process.

Segment Table Entry	- an item in a segment table. Each segment table entry is a segment descriptor.
Segment Table Word	- an entry in a segment table word array pointing to the segment tables of a process. The array is pointed to by one of two address space words in the process control block.
Semaphore	- data structures used to control communication between processes.
Semaphore Block	- a data block containing only semaphore descriptor declarations.
Stack	- a mechanism that accepts stores, and allows retrieval of data on a last-in-first-out basis. Comprised of a number of contiguous portions called stack frames.
Stack Active Area	- the portion of the current stack frame that holds a pointer to the saving area, the caller's instruction counter, parameter space and local variables—in short, the data of the current frame most likely to be useful to execution of the called procedure.
Stack Base Words	- a group of three words in a process control block which contain the segmented address of the stack segments for rings 0, 1, and 2 associated with the process.
Stack Frame	- a contiguous part of the stack segment holding the data stored for a single procedure call.

Stack Operations	- the push-down and pop-up operations of storing and retrieving information to/from the stack. In practice, manipulation of the pointers used to reference the information stored in the stack.
Stack Overflow	- the action of attempting to store more information into the stack then the available space can hold. This is detected by hardware.
Stack Register	- see T register.
Stack Segment	- the segment of a process that contains its stack.
Static Linking	- an intermediate step between compiling a source procedure and executing the object form of that procedure; linking resolves the external references made to and from that procedure.
Status Register	- an 8-bit register which specifies the status of the procedure currently being executed.
Swapping	- releasing the main memory space occupied by a segment by writing the segment out to secondary storage. The actual write to secondary storage may be suppressed if the segment has not been modified since the last time it was read in.
Synchronization	- having a fixed relationship in time; synchronized execution of two or more processes.
System Base	- a fixed area in main memory which acts as a root for all information structures.

Task	- the smallest unit of user-defined work consisting only of a single stream of non-concurrent instructions.
T Register	- a software visible hardware register which contains the segmented address of the top of the procedure stack.
Unit Record Controller	- the elements of the peripheral subsystem related to unit record devices such as card equipment, paper tape equipment, and line printers.
User Process Group	- the internal representation of any job step, as opposed to a system process group which exists independently of any job.
Vacant Indicator	- a hardware structure that indicates that the process which had control of the CP has suspended itself i.e. no CJP in the CPU. Located in Auxiliary Register, of the ALU.
Virtual Memory	- an addressing concept that allows a programmer to code without regard to physical memory size; virtual memory manager automatically manages the exchange of segments between main memory and secondary storage.
V-Operation	- a machine instruction intended for interprocess signaling the completion of an event. A V-OP operates on a designated semaphore.

ABBREVIATIONS

ACU	- address control unit
ALU	- arithmetic and logic unit
AS	- address syllable
ASW	- address space word
BAR	- boundary address register
BCD	- binary coded decimal
BR	- base register
CCE	- channel command entry
CCU	- channel control unit
CMRN	- the callers maximum ring number
CCW	- channel command word
CET	- current state entry time
CIA	- control store interface adapter
CJP	- currently executing process
CPU	- central process unit
CU	- compilation unit
CSU	- control store unit
D	- displacement
DMU	- data management unit
DA	- device adapter
EAR	- effective address ring
EXW	- exception word
GR	- general register
GTW	- G-table word
IC	- instruction counter
IFU	- instruction fetch unit
I/O	- input/output
IOC	- input/output controller
IPQW	- internal process queue word
IR	- index register
ITBB	- indirect to base register
ITS	- indirect to segment

J	- job
JCL	- job control language
JTW	- J-table word
LCT	- logical channel table
LIFO	- last-in-first-out
LSU	- local store memory or scratch pad memory
MBZ	- must be zero
MOS	- metal oxide semiconductor
MAXR	- the maximum ring number at which a procedure may execute MAXR is found in the segment descriptor of SEG_{EP}.
MSC	- mass storage controller
MTC	- magnetic tape controller
NFS	- non-functional status
NPRN	- new process ring number
NJP	- new process replacing the currently executing process
PCB	- process control block
PCT	- physical channel table
PCU	- peripheral control unit
PL/I	- programming language/one
PMW	- process main word
Q/PR/RDY	- queue of processes ready
RD	- read ring
RHU	- reserved for hardware use
RPW	- running process word
RSU	- reserved for software use
RTA	- ready time accounting
RTO	- residual time out
SBW	- stack base word
SEG	- segment number
SKW	- stack word
SEG_{EP}	- the segment which contains the entry point as found in the procedure descriptor

SEG_{PD}	- the segment which contains the procedure descriptor
SR	- scientific register
SRA	- segment relative address
STE	- segment table entry
STN	- segment table number
STR	- status register
STW	- segment table word
STWA	- segment table word array
T	- T-register (stack register)
URC	- unit record controller
WR	- write ring
WTA	- waiting time accounting

Having shown and described one embodiment of the invention, those skilled in the art will realize that many variations and modifications can be made to produce the described invention and still be within the spirit and scope of the claimed invention.

What is claimed is:

* * *

For Claims 3, 5 and 6, see Rejected Claims 3, 5, and 6.

DECLARATION AND POWER OF ATTORNEY (use with sole or joint patent applications accompanying or not accompanying the application papers)

(X) the undersigned declare that the information set forth above and all facts stated herein are true; that (X) he believes that (X) he is the original first and (X) joint inventor of the invention described and claimed in the specification (X) attached hereto or (__) of application Serial Number _____ filed _____; that he has reviewed and understands the contents of the above-identified specification, including the claims, as amended by any amendment referred to above; that he does not know and does not believe that the same was ever known or used in the United States before (X) his invention thereof or patented or described in any printed publication in any country before (X) his invention thereof or more than one year prior to such application; that said invention has not been in public use or on sale in the United States for more than one year prior to the date of this application; that said invention has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States on an application filed by (X) him or (X) his legal representatives or assigns more than twelve months prior to the date of this application; that he does not know and does not believe that the same was ever known or used in the United States before (X) his invention thereof or patented or described in any printed publication in any country before (X) his invention thereof or more than one year prior to the date of this application or in public use or on sale in the United States more than one year prior to the date of this application and that said invention has not been patented in any country foreign to the United States on an application filed by (X) him or (X) his legal representatives or assigns more than twelve months prior to the date of this application; that he does not know and does not believe that any person, firm or corporation has any right, title or interest in the invention except as described in this declaration; that no application for patent or inventor's certificate on said invention has been filed by (X) him or (X) his legal representatives or assigns in any country foreign to the United States except as follows:

20	Bradley	John	J.
0	Garches	France	U.S.A.
1	63 rue du 19 Janvier 92380	Garches	France
20	Franklin	Benjamin	S.
0	Cambridge	Massachusetts	U.S.A.
2	11 Peabody Terrace Apt. #2001	Cambridge	Massachusetts 02138
20			
0			
3			

(X) Further declared that all statements made herein of (X) his own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

SIGNATURE(S) OF APPLICANT(S) (X) HEREIN *	DATE OF SIGNATURE
John J. Bradley	8 April 75

301 TITLE OF INVENTION: Switch System Base Mechanism

600 CORRESPONDENCE ADDRESS

( ) Specific reference to any prior pending application(s) in which this application is a continuation, continuation-in-part or division of, and claiming benefit under 35 USC 120

(X) Hereby appoint the following attorney(s) to prosecute this application and transact all business in the Patent Office connected therewith

NAME	REGISTRATION NUMBER	ASSOCIATION NUMBER
Nicholas Prasinos Ronald T. Reiling	24,791 20,607	(X)

Send Correspondence to: Ronald T. Reiling, Honeywell Information Systems Inc., 200 Smith Street, Waltham, Massachusetts 02154, (617) 890-8400, Ext. 22126

(X) Further declares that all statements made herein of (X) his own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

SIGNATURE(S) OF APPLICANT(S) (X) HEREIN * | DATE OF SIGNATURE

John J. Bradley | 8 April 75

(X) * See attached sheet

WAS ORIGINALLY FI...

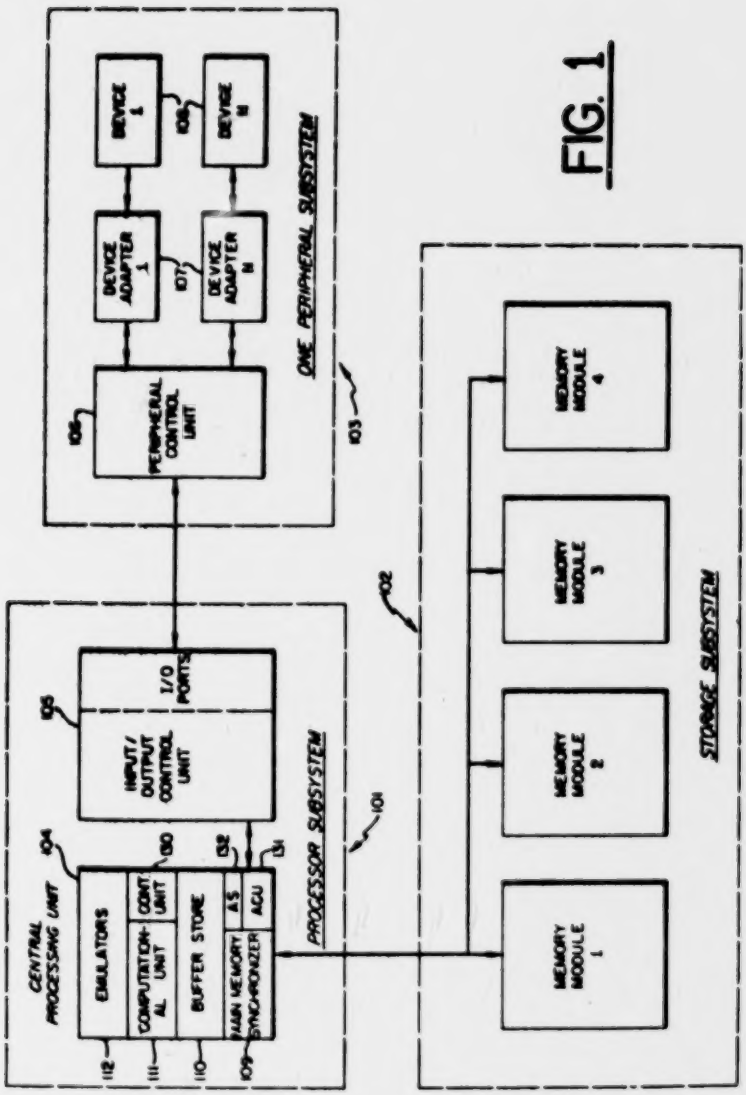


FIG. 1

570331

POINT OF DRAWING AS ORIGINALLY FILED

FIG. 2

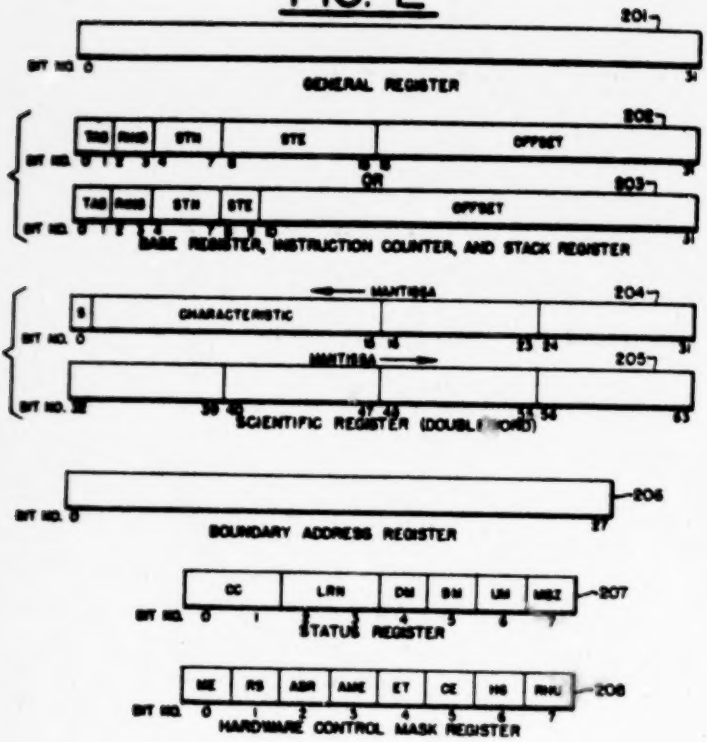


FIG. 3

LEGEND FOR RESERVED AREAS OF STORAGE IN REGISTERS

AME • ACCOUNTING MECHANISM
ABR • AUTO STORAGE RECONFIGURATION
BM • BINARY MASK
CC • CONDITION CODE
CE • CORRECTED ERROR
DM • DECIMAL MASK
ET • ENVIRONMENT TOLERANCE
MB • HARDWARE GATE
LRN • LAST RING NUMBER
MBZ • MUST BE ZERO
ME • MACHINE ERROR
RING • RING NUMBER
RMU • RESERVED FOR HARDWARE USE
RS • RETRY SUCCESSFUL
S • SIGN
STE • SEGMENT TABLE ENTRY
STN • SEGMENT TABLE NUMBER
TAG • DESCRIPTOR TYPE
UM • UNDERFLOW MASK

AS ORIGINALLY FILED

570331

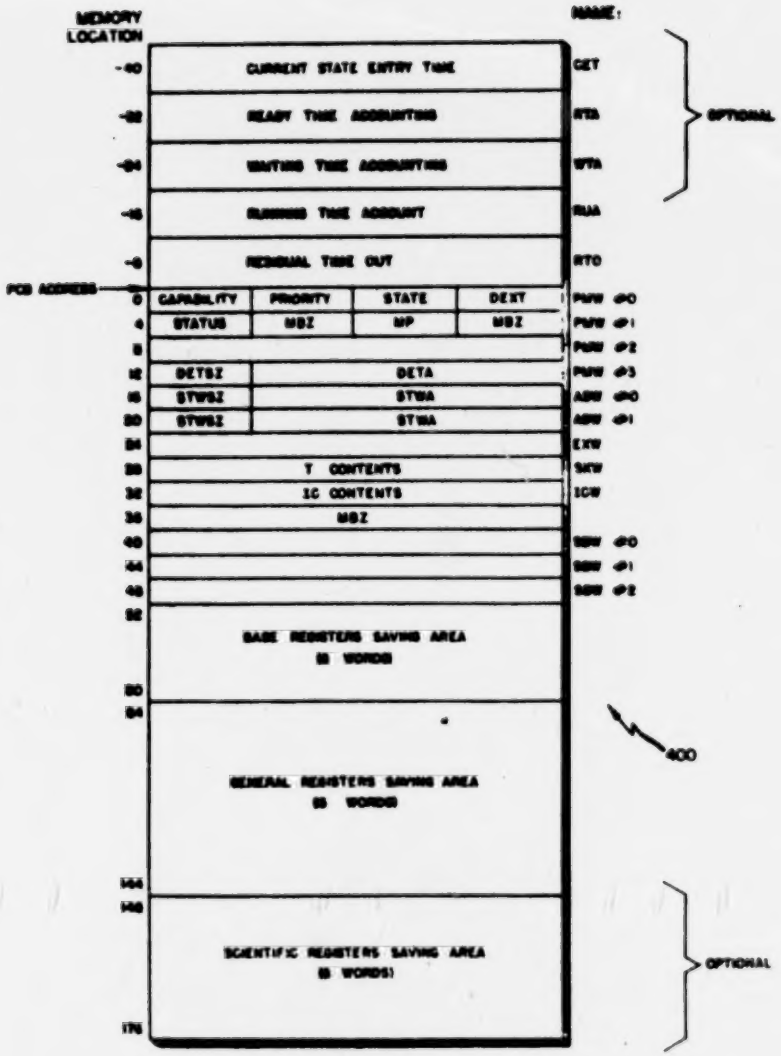


FIG. 4

SIZE OF DRAWING
HAS BEEN ORIGINALLY PLACED

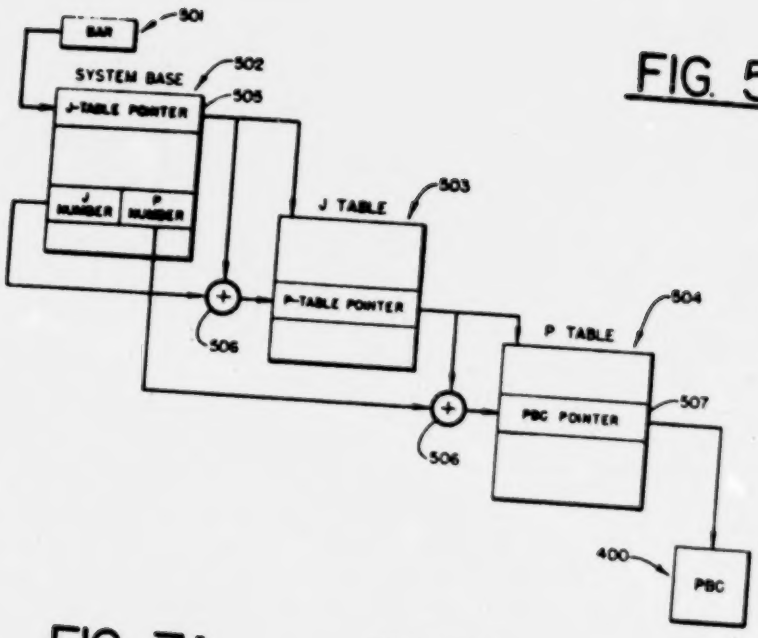
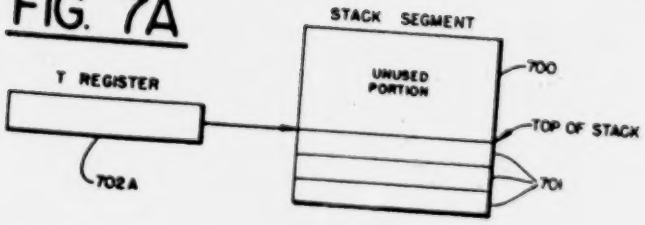


FIG. 5

FIG. 7A



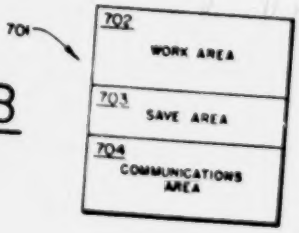
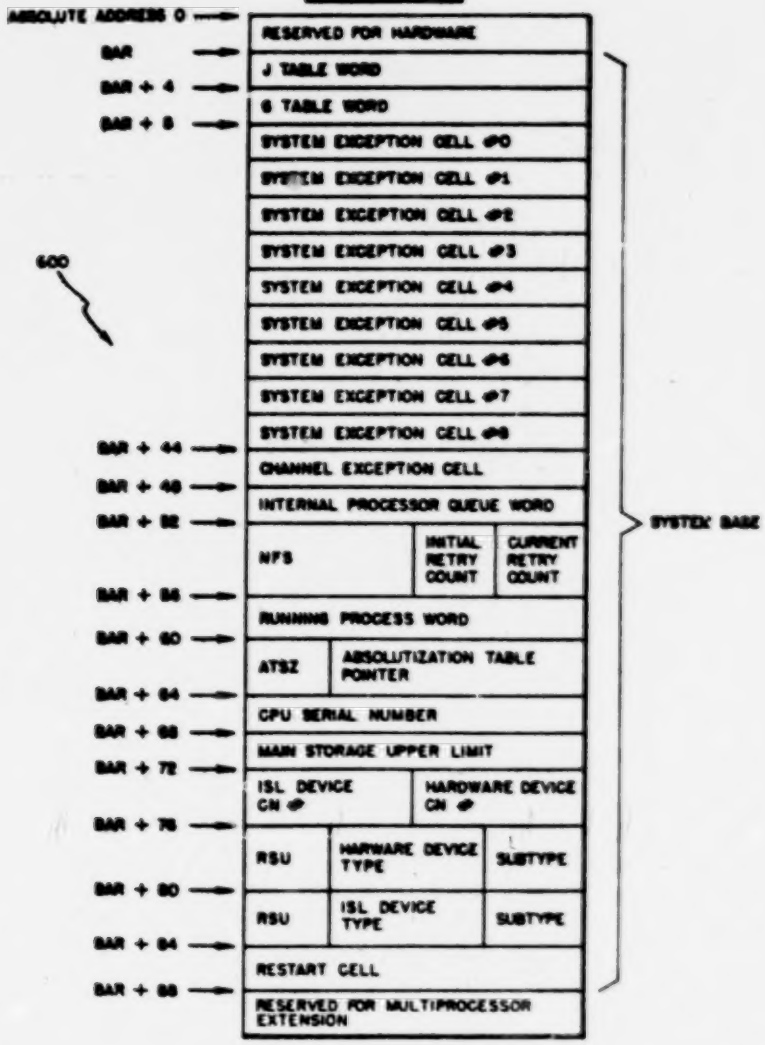


FIG. 7B

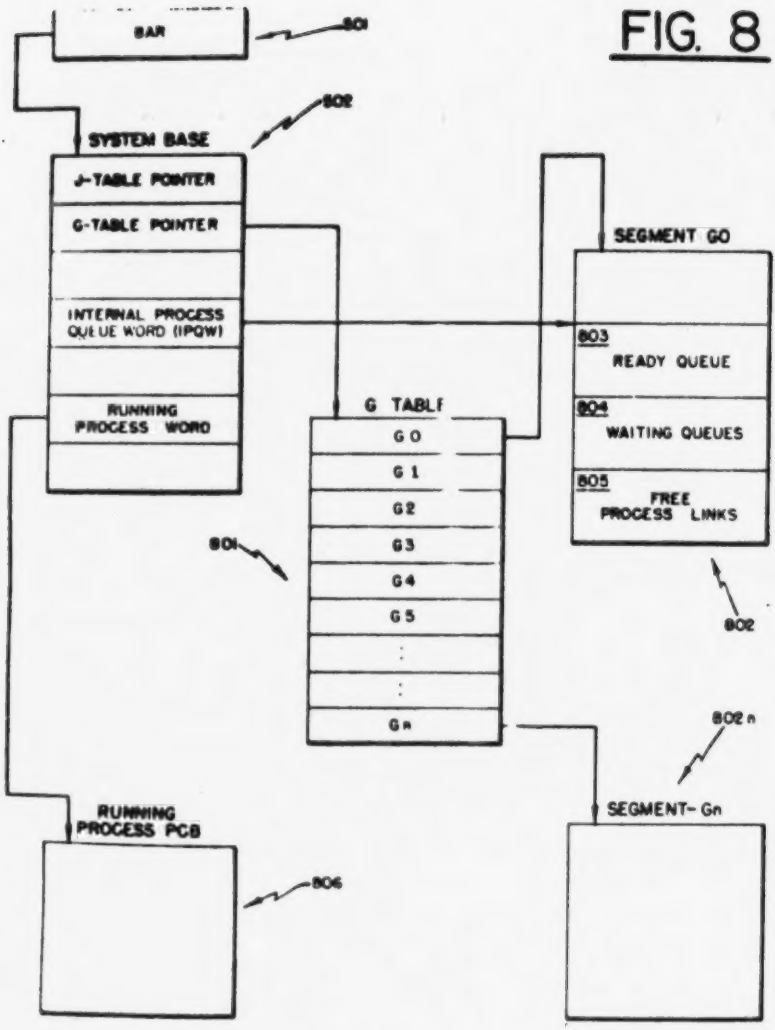
AS ORIGINALLY FILED

FIG. 6



.46 ORIGINALITY

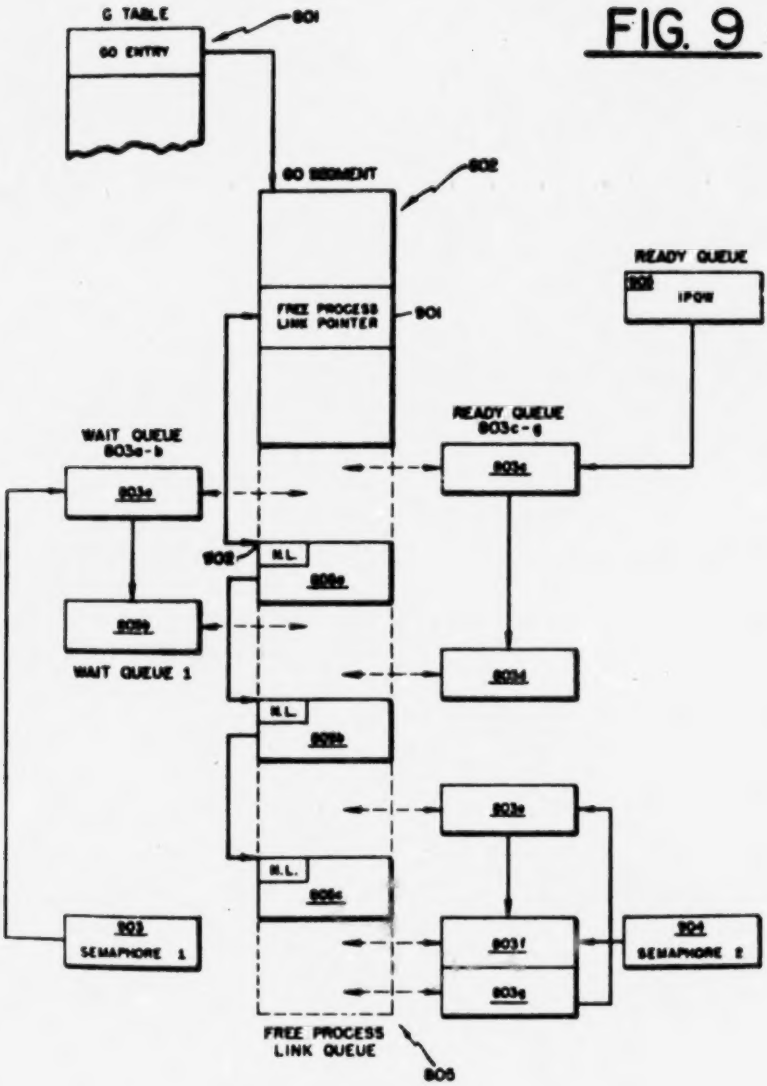
FIG. 8



AS ORIGINALLY FILED

570371

FIG. 9



AS ORIGINALLY FILED

570331

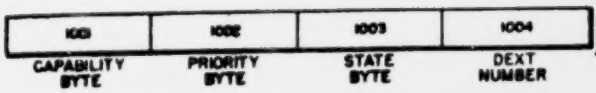


FIG. 10a

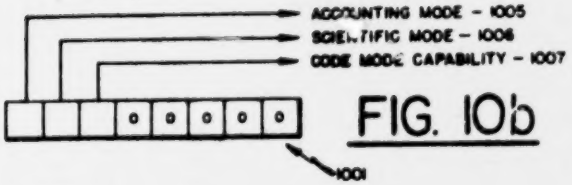


FIG. 10b

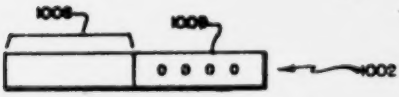


FIG. 10c

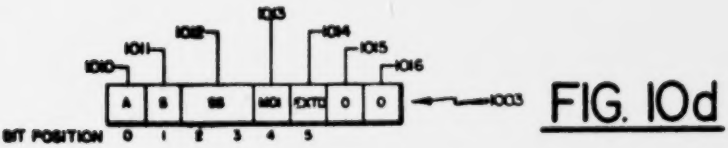


FIG. 10d

FIG. 10e

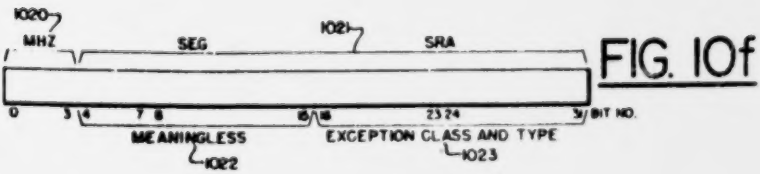


FIG. 10f

49 ORIGINALLY ...
57[illegible]11

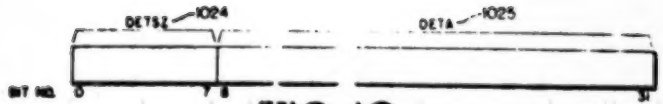


FIG. 10g

FIG. 10h

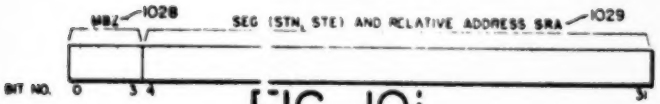


FIG. 10i

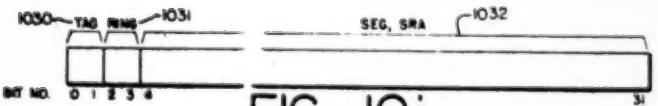


FIG. 10j

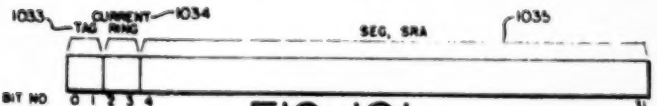


FIG. 10k

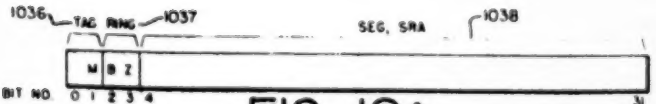


FIG. 10l

AS ORIGINALLY

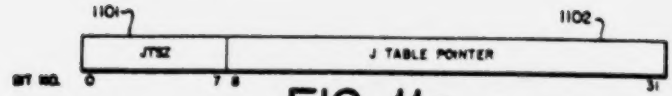


FIG. 11a

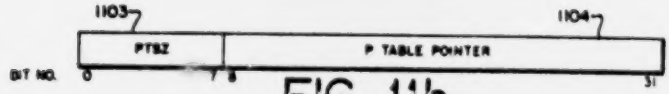


FIG. 11b

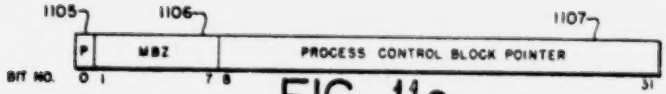


FIG. 11c

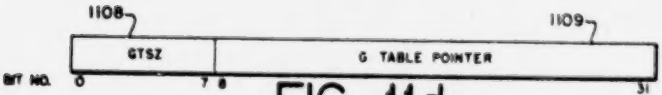


FIG. 11d

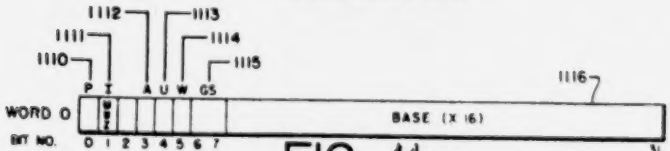


FIG. 11e

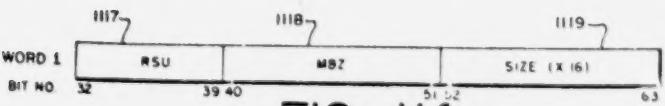


FIG. 11f

...AS ORIGINALLY FI...

570371

FIG. 11g

FIG. 11h

FIG. 11i

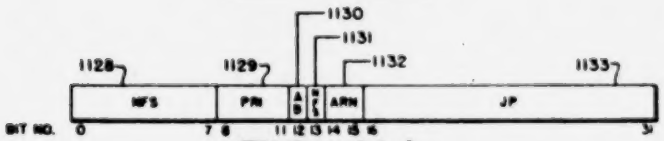


FIG. 11j

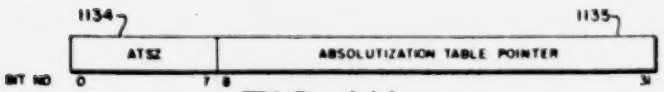


FIG. 11k

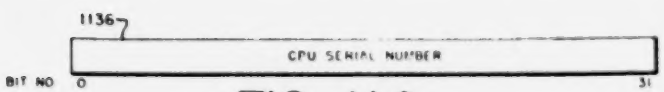


FIG. 11l

PRINT OF DRAWING
AS ORIGINALLY FILED

57:371

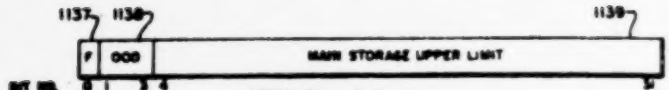


FIG. 11m

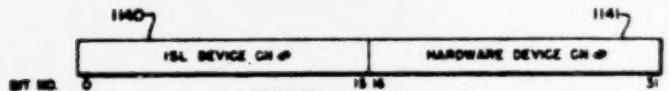


FIG. 11n

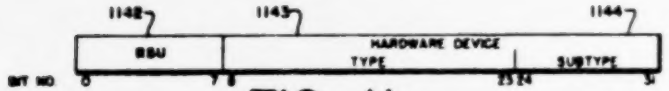


FIG. 11o

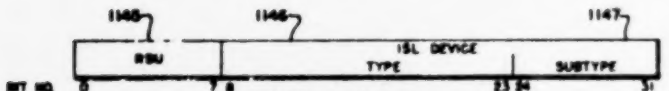


FIG. 11p

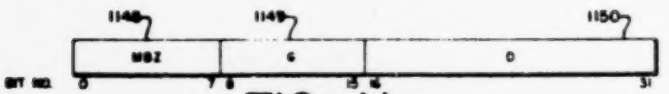


FIG. 11q

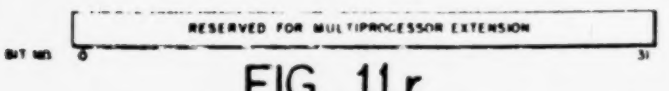
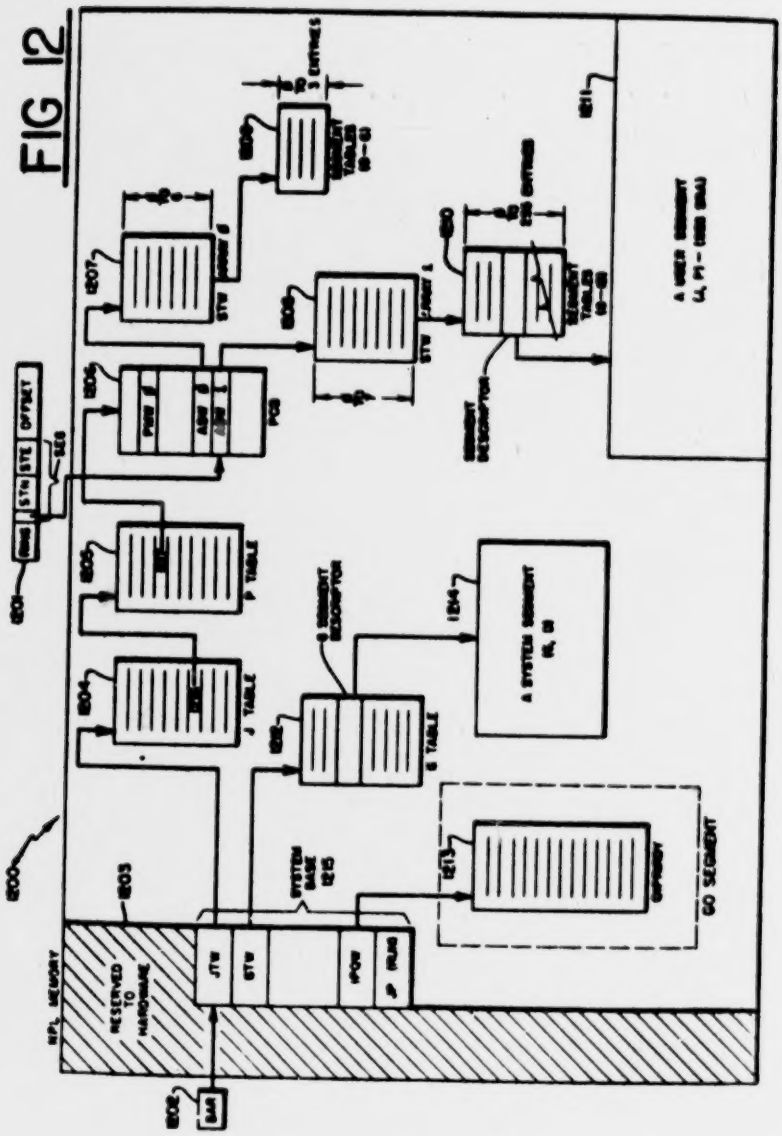


FIG. 11r

FILED WITH DRAWINGS
AS ORIGINALLY FILED

57C3'11

FIG 12



PRINT OF DRAWING AS ORIGINALLY FILED

570371

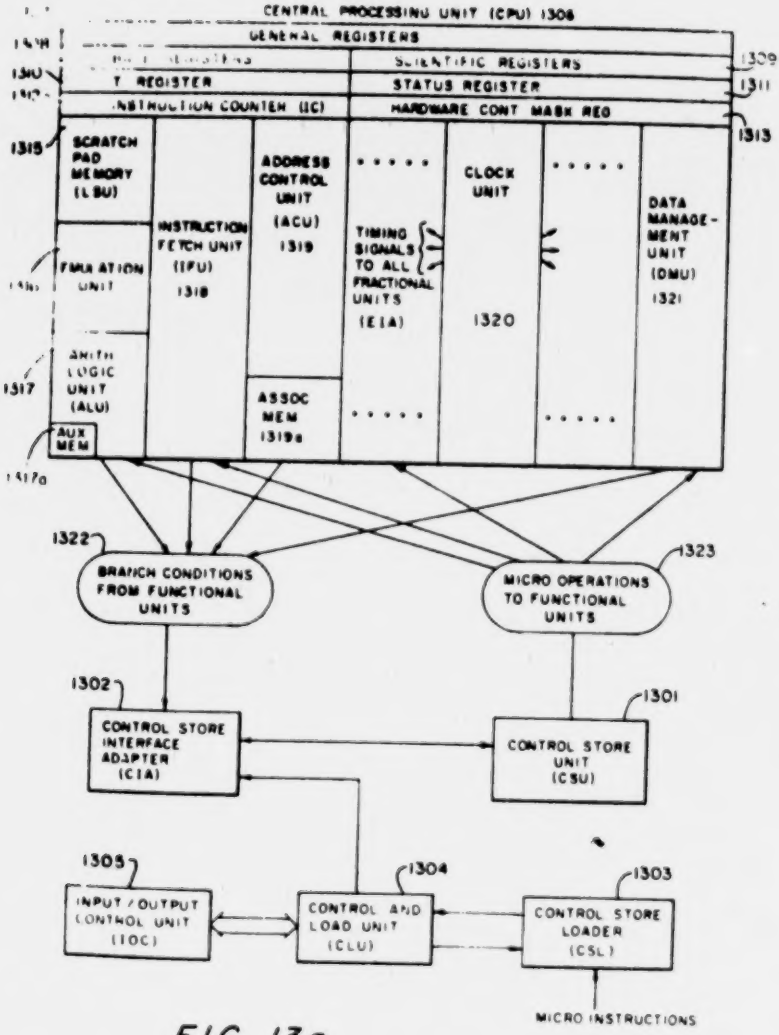
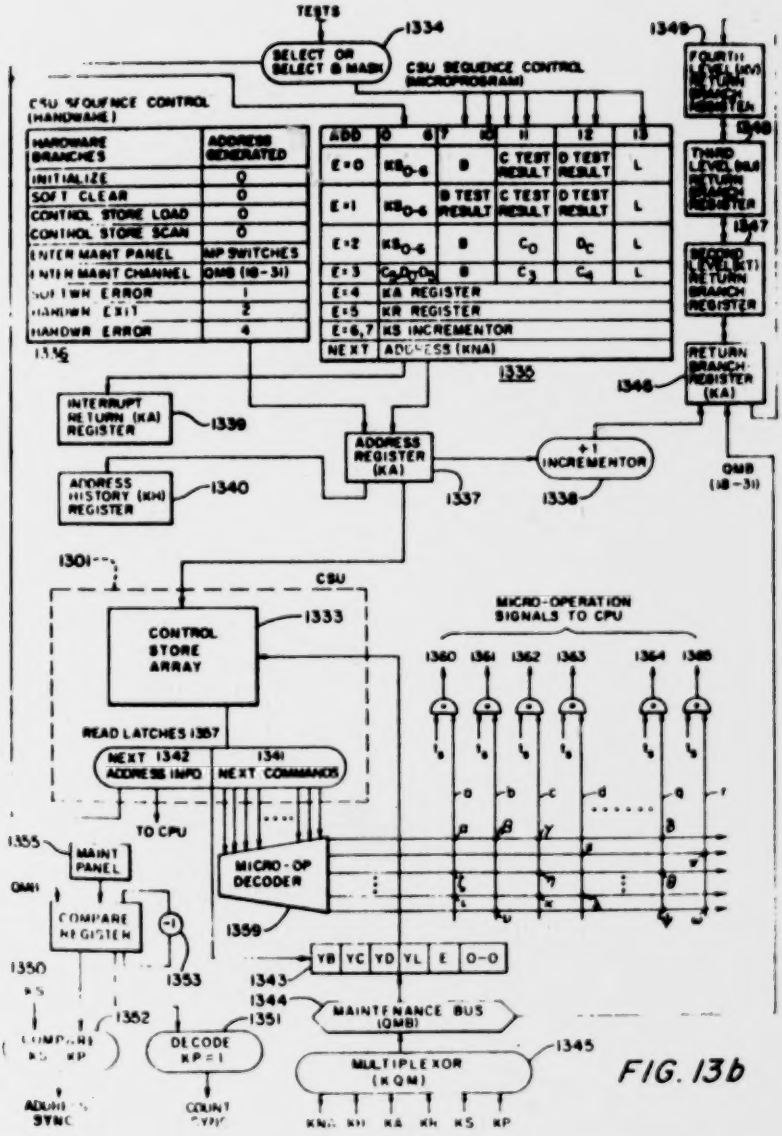


FIG. 13a

PART OF DRAWING AS ORIGINALLY FILED

07C331



570331

PRINT OF DRAWING AS ORIGINALLY FILED

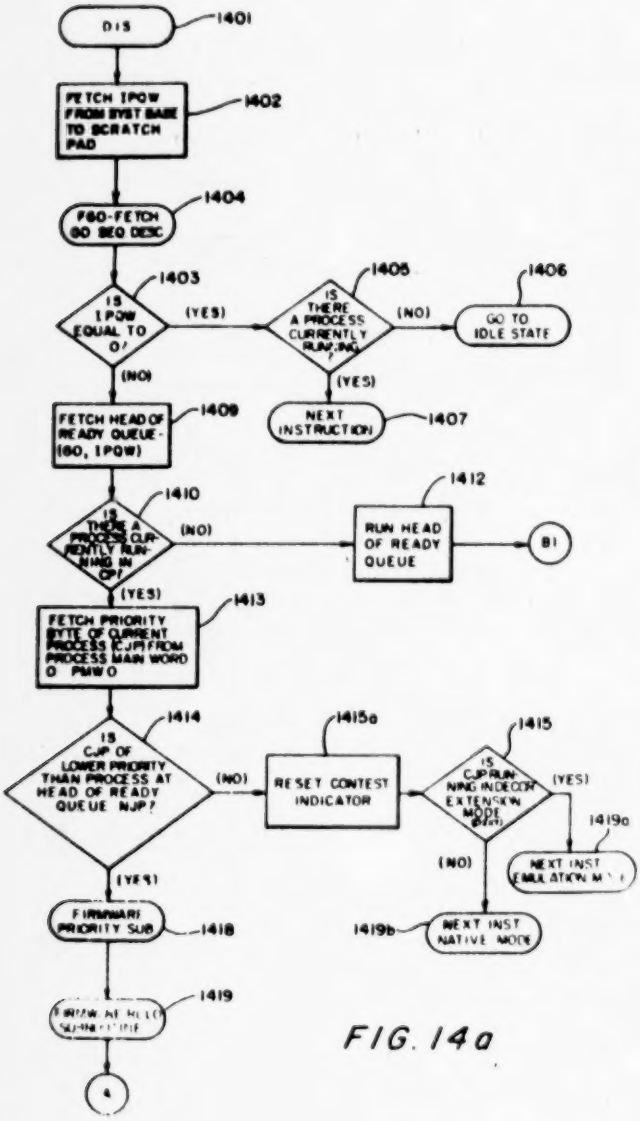


FIG. 14a

PART OF DRAWING
WAS ORIGINALLY FILED

57C331

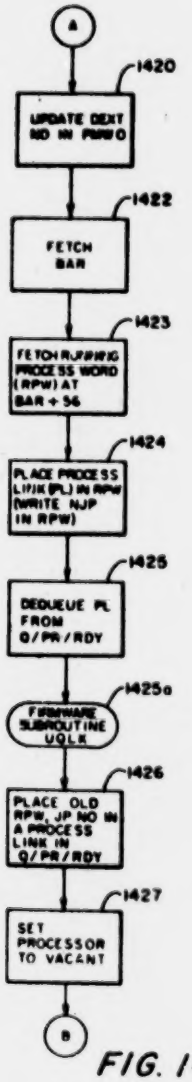


FIG. 14b

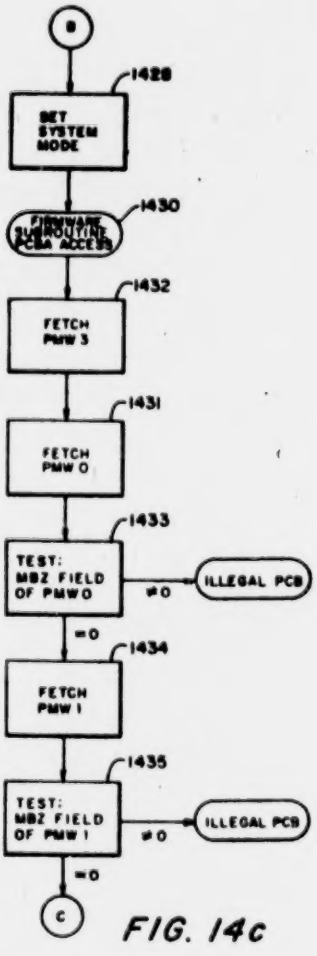


FIG. 14c

PRINT OF DRAWING AS ORIGINALLY FILED

57C371

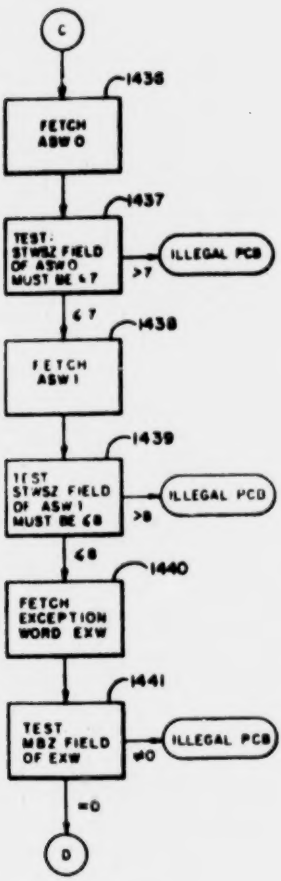


FIG. 14d

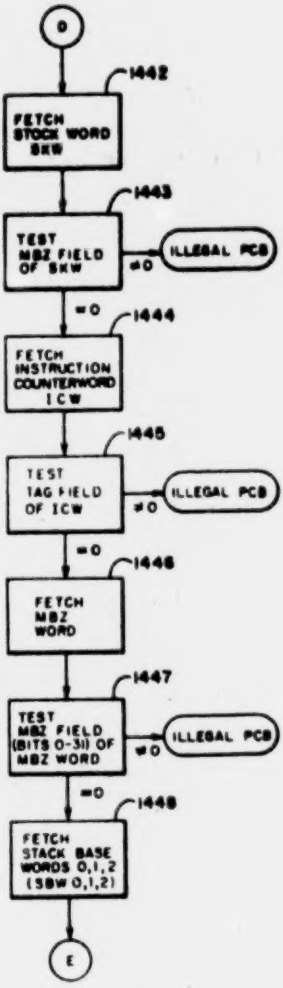


FIG. 14e

TITLE OF DRAWING
WAS ORIGINALLY FILED

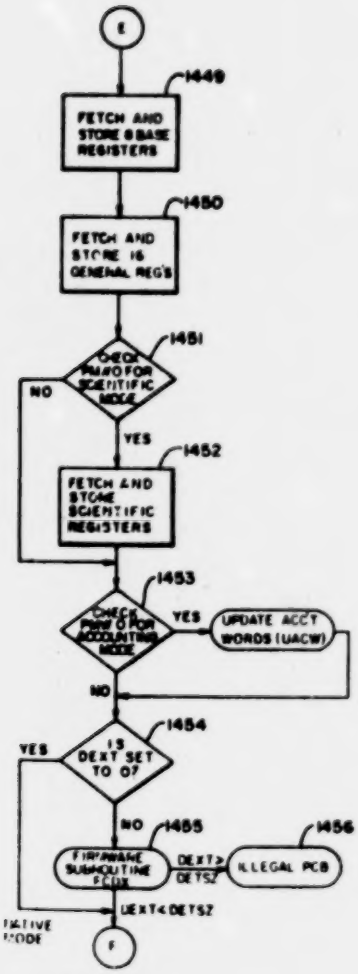


FIG. 14f

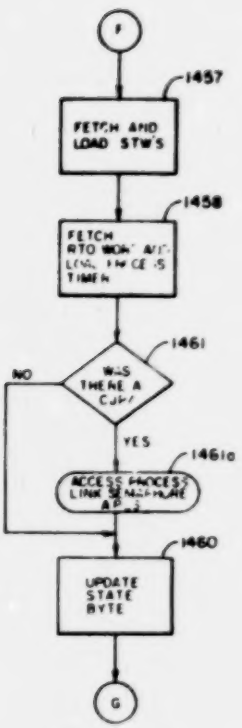


FIG. 14g

WAS ORIGINALLY FILED

570371

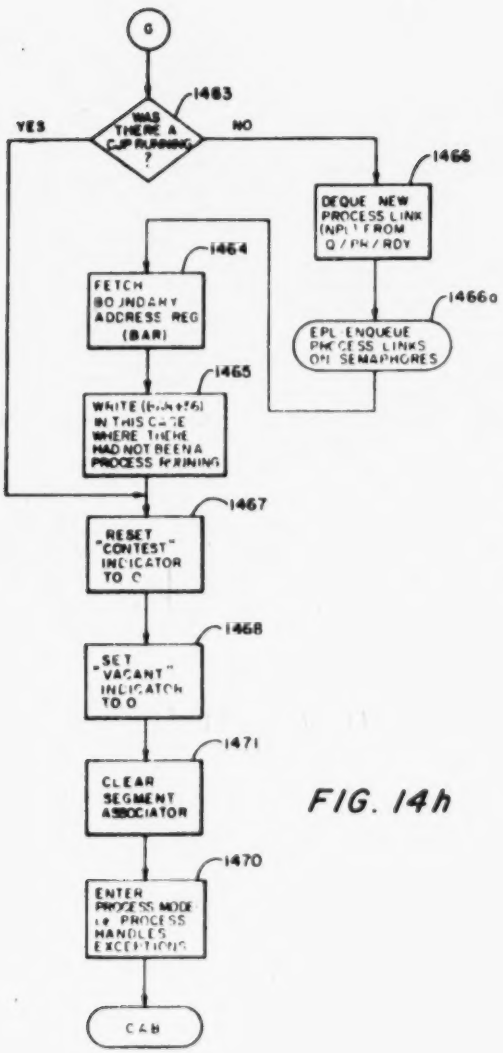


FIG. 14h

[illegible] OF DRAWING
WAS ORIGINALLY FILED

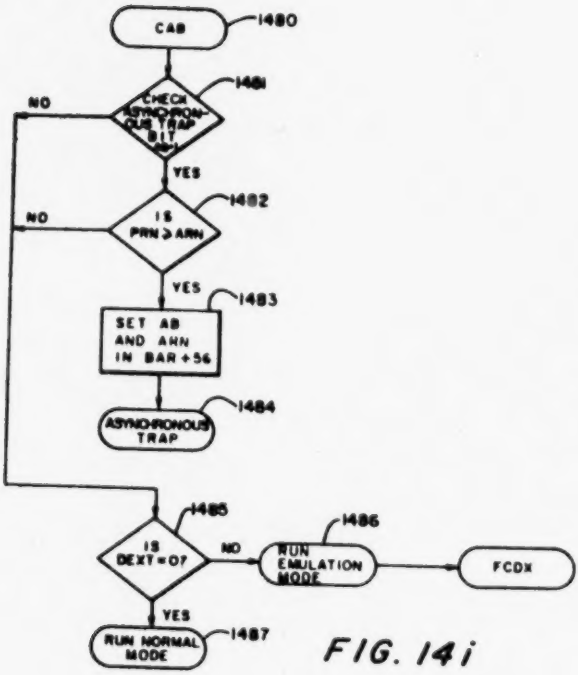


FIG. 14i

MAIN FIELD USES	SEQUENCE TYPE	BRANCHING AND / OR MICRO-OPS					CONSTANT GENERATION AND DESIGNATION	DATA TO BUS		MICRO-OPS	CHECKING
FIELD NAME	E	A	B	C	D	L	K	QA	QB	F	P
BITS	3	6	4	6	6	1		4	4		
		23					14	8		32	4

1326 1327 1328 1329 1330 1331 1325

FIG. 13c

570331

PRINT OF DRAWING
AS ORIGINALLY FILED

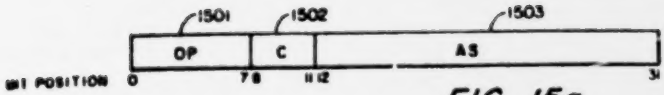


FIG. 15a

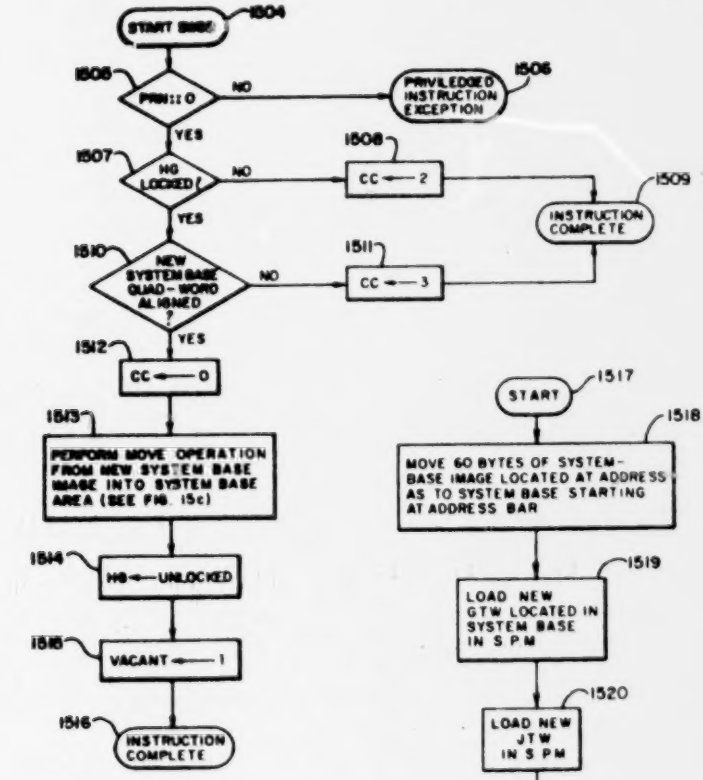


FIG. 15b

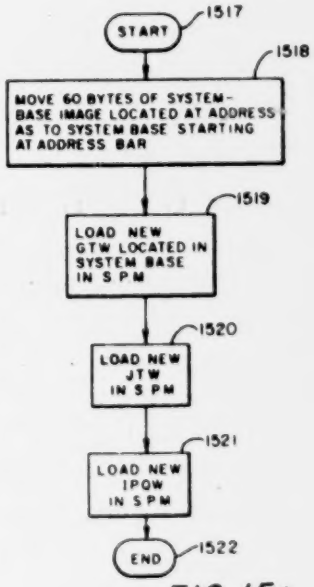


FIG. 15c

REJECTED CLAIMS.

1. In a multiprogramming computer system having a main memory, a central processing unit (CPU) coupled to said main memory, said (CPU) controlling the state of a plurality of groups of processes being in a running, ready, wait or suspended state, said computer system also having scratchpad registers being accessible to an operating system for controlling said multiprogramming computer system, a data structure for storing coded signals for communicating between said processes and said operating system, and said scratchpad registers, said data structure comprising:

(a) first means in said data structure and communicating with said operating system for storing coded signals indicative of an address for a selected one of said processes;

(b) second means in said first means for storing coded signals indicating priority of said selected one of said processes in relation to others of said processes for obtaining control of said CPU when ready;

(c) third means in said data structure and communicating with said operating system, for storing coded signals indicative of an address for a selected one of said plurality of groups of processes, and,

(d) fourth means coupled to said data structure and said scratchpad registers, for generating signals causing the changing of information in said date structure and said scratchpad registers.

2. In combination with a communication center located in a predetermined area of a computer main memory the computer having a central processing unit (CPU) for executing plurality of processes that compete with each other for control of said CPU, operating system means coupled for providing information to said communication center, and a scratch pad memory in said CPU which stores, under control of said operating system means, portions of information

located in said communication center, an improved system controlled by one of said processes for changing information in said communication area and in said scratchpad memory, said improvement comprising:

(a) first means for generating signals causing the creation of an image of said communication center at another location in said main memory;

(b) second means responsive to said operating system means for generating signals causing a change of the information in said image, resulting in a new image; and,

(c) third means, coupled for being responsive to said first and second means and to one of said processes for generating signals for causing a transfer of the new image to said communication center and to said scratchpad memory.

3. The system as recited in Claim 2 wherein said means means is a switch system base instruction in firmware.

4. The system as recited in Claim 3 further including an arithmetic and logic unit having an auxiliary memory said arithmetic and logic unit coupled for communicating with said first, second and third means.

5. The system as recited in Claim 4 wherein said auxiliary memory includes a hardware gate which has a locked and unlocked state said unlocked state permitting internal interrupts or external exceptions, and said locked state not permitting internal interrupts or extral exceptions.

6. The system as recited in Claim 5 wherein said hardware gate is locked by said switch system base instruction when executing and unlocked by said switch system base instruction when it terminates execution.

FILED
FEB 7 1980

No. 79-855

In the Supreme Court of the United States

OCTOBER TERM, 1979

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS, PETITIONER

v.

JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN

ON PETITION FOR A WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

REPLY MEMORANDUM FOR THE PETITIONER

WADE H. MCCREE, JR.
Solicitor General
Department of Justice
Washington, D.C. 20530

In the Supreme Court of the United States

OCTOBER TERM, 1979

No. 79-855

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS, PETITIONER

v.

JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN

ON PETITION FOR A WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

REPLY MEMORANDUM FOR THE PETITIONER

Respondents contend, as they did below (see Pet. 8 n. 12), that they have created a novel arrangement of computer hardware (Br. in Opp. 14-15, 19); they argue that *Parker* v. *Flook*, 437 U.S. 584 (1978), is distinguishable since the process in *Flook*, aside from the mathematical algorithm, was admittedly old (Br. in Opp. 15-16).

The premise of this argument simply renews respondents' disagreement with the Board of Appeals' findings, which the CCPA declined to disturb. Although respondents characterize their claim as a novel machine comprised of an arrangement of hardware including a firmware element, what matters is whether that description masks a claim to a machine that is old in the art, whose only novelty is an unpatentable algorithm

embodied in the firmware (Pet. 9-10).[1] The Board, affirming the examiner on this point, found that "save for the microprogramming," Bradley's arrangement of hardware was old in the art (Pet. App. 26a-27a; see Pet. 8). The CCPA, while noting respondents' challenge to this determination, left standing the Board's finding on this factual issue (Pet. App. 13a). Given the prior art as set out in the Patent Office record, the court could hardly have done otherwise. Respondents incorporated by reference in their application an earlier application by Claude Carre and others that discloses an arrangement of hardware identical to the arrangement claimed in respondents' application (Pet. App. 43a-44a, 52a-53a). Respondents now concede as much (compare Br. in Opp. 12 n.22 with Pet. 8-9 n.12).[2]

The program in firmware is thus the only novel aspect of respondents' invention. As we have shown (Pet. 11-12), however, that program is not itself patentable, and respondents accordingly do not have a patentable invention under the analysis required by *Parker* v. *Flook*, *supra*, 437 U.S. at 594-596. Respondents contend (Br. in Opp. 11, 17) that petitioner's interpretation of *Flook* is incorrect, and that the CCPA here and in other cases has correctly followed this Court's teaching in that

---

[1]Thus respondents' assertion that it is conceded that their claimed invention "is a machine" (Br. in Opp. 3, 19) is beside the point.

[2]Moreover, in their application respondents noted that their invention could be practiced with "computer processors including microprogrammed control units" disclosed in specified issued patents (R. 126). This language strongly suggests that respondents recognized in their application that their inventive contribution was in the additional matter not contained in the Carre application—the microprogram—and not in the hardware with which it was implemented.

case (Br. in Opp. 17-18 & nn.30, 31). That is the issue presented in this case and in *Diamond* v. *Diehr*, No. 79-1112, and it is that question that merits review by this Court.

For the reasons stated herein and in the petition, the petition for a writ of certiorari should be granted.

Respectfully submitted.

WADE H. MCCREE, JR.
Solicitor General

FEBRUARY 1980

Supreme Court, U.S.
FILED
JUN 10 1980
MICHAEL RODAK, JR., CLERK

No. 79-855

In the Supreme Court of the United States

OCTOBER TERM, 1979

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS, PETITIONER

v.

JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

BRIEF FOR THE PETITIONER

WADE H. MCCREE, JR.
Solicitor General

SANFORD M. LITVACK
Assistant Attorney General

LAWRENCE G. WALLACE
Deputy Solicitor General

HARRIET S. SHAPIRO
Assistant to the Solicitor General

ROBERT B. NICHOLSON
FREDERIC FREILICHER
Attorneys
Department of Justice
Washington, D.C. 20530

JOSEPH F. NAKAMURA
Solicitor

THOMAS E. LYNCH
Associate Solicitor
United States Patent and Trademark Office
Washington, D.C. 20231

INDEX

	Page
Opinions below	1
Jurisdiction	1
Question presented	2
Statute involved	2
Statement	2
Summary of argument	8
Argument:	
I. Respondents' claims lack patentable subject matter	9
A. Computer Programs are not patentable	11
B. Algorithms codes in firmware are not patentable	19
II. It is for Congress to determine whether, and the extent to which, computer programs ought to be protect	22
Conclusion	25

CITATIONS

TABLE OF AUTHORITIES

Cases:	
Berardini v. *Tocci*, 190 F. 329, aff'd on the opinion below, 200 F. 1021	13
Cochran v. *Diener*, 94 U.S. 780	14
Corning v. *Burden*, 56 U.S. (15 How.) 252	17
Detroit Gasket & Mfg. Co. v. *Fitzgerald Mfg. Co.*, 89 F.2d 178	18
Expanded Metal Co. v. *Bradford*, 214 U.S. 366	17
Freeman, In re, 573 F.2d 1237	22

Cases (continued):

Page

Funk Brothers Seed Co. v. *Kalo Incoculant Co.*, 333 U.S. 127 16
General Electric Co. v. *Jewel Incandescent Lamp Co.*, 326 U.S. 242 18
Ghiron, In re, 442 F.2d 985 3
Gottschalk v. *Benson*, 409 U.S. 63 ..... 6, 8, 9, 16, 19–20, 22, 23
Graham v. *John Deere Co.*, 383 U.S. 1 ... 11, 18
Honeywell, Inc. v. *Lithonia Lighting, Inc.*, 317 F. Supp. 406 ..... 2, 3
Hotel Security Checking Co. v. *Lorraine Co.*, 160 F.2d 467 13
Johnson, In re, 589 F.2d 1070 15
Johnston, In re, 502 F.2d 765 22
Joseph E. Seagram & Sons v. *Marzall* 180 F.2d 26 ..... 10, 13
Kothny, In re, 96 F.2d 289 21
Loews Drive-In Theatres, Inc. v. *Park-In Theatres, Inc.*, 174 F.2d 547 13
Maucorbs, In re, 609 F.2d 481 19
Murphey v. *American Ry. Express Co.*, 24 F.2d 318 21
Newton Steel Co. v. *Surface Combustion Co.*, 75 F.2d 305 17
O'Reilly v. *Morse*, 56 U.S. (15 How.) 62 .. 9, 16
Parker v. *Flook*, 437 U.S. 584 ..... 7, 8, 9, 12, 15, 16, 19, 22, 23
Patton, In re, 127 F.2d 324 13
Risdon Iron & Locomotive Works v. *Medart*, 158 U.S. 68 17
Roberts v. *Ryer*, 91 U.S. 150 18
Rubber-Tip Pencil Co. v. *Howard*, 87 U.S. (20 Wall) 498 16
Russell, In re, 48 F.2d 668 20–21

Cases: (continued):

Page

Sherwood, In re, 613 F.2d 809, petition for cert. pending 14
Vapor Car Heating Co. v. *Gold Car Heating & Lighting Co.*, 7 F.2d 284 17
Wadman, In re, 94 F.2d 993 17
Wait, In re, 73 F.2d 982 13
Waldbaum, In re, 457 F.2d 997 3
Walter, In re, 205 U.S.P.Q. 397 ..... 21, 22
Westinghouse v. *Boyden Power Brake Co.*, 170 U.S. 537 17
Wier v. *Coe*, 33 F.Supp. 142, aff'd per curiam, 122 F.2d 959 .....
Wyeth v. *Stone*, 30 F. Cas. 723 16
Zierden, In re, 411 F.2d 1325 18

Statutes:

35 U.S.C. 101 ..... 2, 5, 21
35 U.S.C. 112 11
17 U.S.C. (Supp. II) 102 24

Miscellaneous:

ABA Patent, Trademark and Copyright Law Section, *1972 Committee Reports* (1973) 23
ABA Patent, Trademark and Copyright Law Section, *1972 Summary of Proceeding* (1973) 23
Breyer, *The Uneasy Case For Copyright: A Study Of Copyright In Books, Photocopies And Computer Programs*, 84 Harv. L. Rev. 281 (1970) 12
Cummingham & Smith, "Computer Services: A Menu of Options," *Datamation* (May 25, 1979) 3

Miscellaneous (continued):

Page

Final Report of the National Commission on New Technological Uses of Copyrighted Works (1979) 24

IEEE Computer Societ Technical Committee on Microprogramming, *Microprogramming Definitions*, 7 SIG Micro Newsletter 7 (Dec. 1976) 4

P. Jordain & M. Breslau, Condensed Computer Encyclopedia (1969) 4

R. Kline, *Digital Computer Design* (1977) . 4, 20

Pope and Pope, *Protection of Proprietary Interests In Computer Software*, 30 Ala. L. Rev. 527 (1979) 24

Ross, *The Patentability of Computer "Firmware,"* 59 J. Pat. Off. Soc'y 731 (1977) 2, 3, 4

C. Sippl & C. Sippl, *Computer Dictionary and Handbook*, (2d ed. 1972) 3, 6, 10–11

"To Promote the Progress of . . . Use Arts," Report of the President's Commission of the Patent System (1966) (reprinted in Subcomm. On Patents, trademarks, and Copyrights, Senate Comm. on the Judiciary, Doc. No. 5, 90th Cong., 1st Sess. (1967) 23

126 Cong. Rec. H2263 (daily ed. Mar. 26, 1980) 24

In the Supreme Court of the United States

OCTOBER TERM, 1979

No. 79-855

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS, PETITIONER

v.

JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

BRIEF FOR THE PETITIONER

OPINIONS BELOW

The opinion of the Court of Customs and Patent Appeals (J.A. 10-20) is reported at 600 F.2d 807. The opinion of the Patent and Trademark Office Board of Appeals (J.A. 193-203) and the opinions of the patent examiner (J.A. 121-125, 150-152, 163-164) are not reported.

JURISDICTION

The judgment of the Court of Customs and Patent Appeals was entered on July 5, 1979 (J.A. 20a). On

October 2, 1979, the Chief Justice extended the time to file a petition for a writ of certiorari to and including December 2, 1979, and the petition was filed on Monday, December 3, 1979. Certiorari was granted on March 17, 1980. The jurisdiction of this Court is invoked under 28 U.S.C. 1256. *Gottschalk* v. *Benson*, 409 U.S. 63 (1972); *Dann* v. *Johnston*, 425 U.S. 219 (1976); *Parker* v. *Flook*, 437 U.S. 584 (1978).

QUESTION PRESENTED

Whether a computer program that regulates the internal operation of a computer is patentable subject matter under 35 U.S.C. 101.

STATUTE INVOLVED

35 U.S.C. 101 provides:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

STATEMENT

On April 21, 1975, John J. Bradley and Benjamin S. Franklin applied for a patent (J.A. 24) on an invention that directs the transfer of data within a computer.[1].

The computer's hardware[2] includes a main memory,

---

[1] The invention is assigned to Honeywell Information Systems, Inc. ("Honeywell") (Br. In Opp. 2).

[2] The physical components of the computer are called "hardware." *Honeywell, Inc.* v. *Lithonia Lighting, Inc.*, 317 F.Supp. 406, 408 (N.D. Ga. 1970); Ross, *The Patentability of Computer "Firmware,"* 59 J. Pat. Off. Soc'y 731, 736 (1977).

which has a system base[3] with which the user of the computer can communicate.[4] In high performance computers, where speed of operation is important, certain information that is part of the system base may be stored in storage components of the computer called "scratchpad registers" (J.A. 11 & n.1). Because these registers are not, however, normally accessible to the user by means of computer programs (J.A. 11), changing the data they contain has been a problem.[5]

Respondents' claimed invention facilitates changing the data in the registers. In addition to certain hardware elements, it employs "firmware" (J.A.11). Firmware consists of a sequence of basic instructions stored as a microprogram in the computer's control

---

[3] The system base stores information used to control the computer in carrying out programmed operations.

[4] A user wanting a general purpose digital computer to perform a certain assigned task communicates with the computer by translating the task into coded instructions that can be read, understood and executed by the computer system. These instructions, called a "program," are written in special programming languages, such as COBOL or FORTRAN. See C. Sippl & C. Sippl, *Computer Dictionary and Handbook* 80, 191 (2d ed. 1972); *In re Ghiron*, 442 F.2d 985, 986 (C.C.P.A. 1971); *In re Waldbaum*, 457 F.2d 997, 998 (C.C.P.A. 1972). Computer programs as well as the information (data) fed into the computer are commonly called "software." *Honeywell, Inc.* v. *Lithonia Lighting, Inc.*, *supra*, 317 F. Supp. at 408; Ross, *supra*, 59 J. Pat. Off. Soc'y at 736. Computer users either write their own programs or obtain them from software vendors. In 1978, user expenditures for software products amounted to $981 million. Cunningham & Smith, "Computer Services: A Menu of Options," *Datamation* 89, 91 (May 25, 1979).

[5] To change the data in the scratchpad registers the user has had to reload the entire system base, a time consuming task, or use software limited to that particular computer model, an option unacceptable to many computer users (J.A. 11).

memory.[6] When activated by software, the instructions cause a particular sequence of computer operations to take place.[7] An instruction in the firmware developed by respondents permits the user to communicate with the scratchpad registers and to switch data back and forth from the registers to the system

---

[6] A microprogram has been defined as a " 'sequence of elementary steps which permits the computer hardware to carry out a program instruction.' " Ross, *supra*, 59 J. Pat. Off. Soc'y at 754, quoting from P. Jordain & M. Breslau, *Condensed Computer Encyclopedia* 318 (1969). Thus, the microprogrammer, like the programmer, uses special programming language to prepare instructions for the computer to carry out in sequence. Ross, *supra*, 59 J. Pat. Off. Soc'y at 765. The microprogram is usually stored in a "control memory," in the instant specification called a "control store" or "control unit" (J.A. 60), physically separate from other computer memories. Ross, *supra*, 59 J. Pat. Off. Soc'y at 758. Early control memories were of the "read only" variety ("ROM") which meant that the microprogram was physically fixed in the control memory by the manufacturer. To change the microprogram the manufacturer had to install a new ROM with the new microprogram. Technical advances made it possible for the user himself to microprogram the control memory (programmable read only memory, "PROM"). Although the original PROMs could be programmed only once, many large computers now have PROMs that permit the user to reprogram the control memory any number of times electrically, without hardware alteration. *Id.* at 760–761; R. Kline, *Digital Computer Design* 333–335 (1977); IEEE Computer Society Technical Committee on Microprogramming, *Microprogramming Definitions*, 7 SIG Micro Newsletter 7, 12 (Dec. 1976).

[7] In respondents' specification the firmware is stored in the "control store unit CSU 1301" (J.A. 60). Typically this unit has a "programmable read-only memory (PROM) mixed with read/write random access store (RAM)" (*id.* at 61) (see note 6, *supra*). Thus while microprograms normally "are loaded from an external source during system initialization and become a permanent control function of the machine," the unit "has the ability to be reloaded and initialized in a manner that provides for a variety of * * * operational modes" (J.A. 60).

base in the main memory (*id.* at 11–13).[8] While respondents' specification describes an arrangement of computer machinery (*id.* at 99–119), the microprogram's use is not limited just to the digital computer so described. "[A]ny control store [control memory] having a ROM" may be "microprogrammed with the flow charts of Figures 15b and 15c" of respondents' specification (*id.* at 141), and respondents make particular reference to two other patented "computer processors" that include "microprogrammed control units with which the instant invention may be practiced" (*id.* at 134; but cf. Br. in. Opp.7 n.13).[9]

The patent examiner rejected the invention as drawn to nonstatutory subject matter under 35 U.S.C. 101 (J.A. 151–152, 163–164).[10] He noted that the hardware arrangement to which the claims re-

---

[8] When the computer user activates what respondents call a "switch system base instruction," this initiates the execution of a microprogram that transfers the data (J.A. 27–28). In respondents' specification, the switch system base instruction is disclosed by the diagram in Figure 15a (*id.* at 78, 120) and the microprogram steps are disclosed in the flow charts of Figures 15b and 15c (*id.* at 79, 120).

[9] As respondents explained in the amendment to their application (J.A. 143), "* * * the present invention is directed to hardware structures controlled by a microprogrammed control system which provides for the generation of [specified] signals * * *."

[10] The examiner's initial rejection (J.A. 121–125) was also based on inadequate disclosure, failure to "particularly point out and distinctly claim the subject matter sought to be patented" (*id.* at 123), and prior art (the Hauck patent), *id.* at 124. Respondents then amended their application, modifying the specification (*id.* at 126–129), offering citations to show where the principal components of their claimed invention had been previously used or identified (*id.* at 129–134), and distinguishing

ferred—"a main memory, central processing unit, and scratchpad registers"—were "well known components" that were "admittedly old in the art * * *" (*id.* at 151-152).[11] He found that the only novel aspect of the invention resided in an algorithm[12] designed to direct the computer to solve the particular problem indicated (*ibid.*).[13] He concluded that this "program implemented algorithm" was not patentable under *Gottschalk* v. *Benson*, 409 U.S. 63 (1972) (*ibid.*).

The Patent and Trademark Office Board of Appeals agreed with the examiner (J.A. 202). It found that "save for the microprogramming" respondents' arrangement of hardware was old in the art (*id.* at 200). It ruled that the fact that respondents' claims were in "apparatus," rather than "methods," format did not

---

the Hauck patent (*id.* at 147-149). The examiner thereupon reaffirmed his rejection of the application on the ground that it was directed to nonstatutory subject matter (*id.* at 150-152).

[11] That admission was contained in the amendment to the application (note 10, *supra*; J.A. 129-134).

[12] In the context of computer technology, an algorithm is a term of art:

> 1. A fixed step-by-step procedure for accomplishing a given result; usually a simplified procedure for solving a complex problem, also a full statement of a finite number of steps. 2. A defined process or set of rules that leads [sic] and assures development of a desired output from a given input. A sequence of formulas and/or algebraic/logical steps to calculate or determine a given task; processing rules.

C. Sippl & C. Sippl, *supra*, at 23.

[13] The examiner further noted in his answer before the Patent and Trademark Office Board of Appeals that microprogrammed control units like respondents' were also old in the art (J.A. 182). Thus, he concluded that the invention "resides in an instruction in firmware which changes, upon execution, information in the system base" (*ibid.*).

make them any less "related to an algorithm" than was the claim in *Benson* (*id.* at 201).[14] Added the Board, "[a] claim for an improved method of calculation, even when tied to a specific end use, is unpatentable" under *Parker* v. *Flook*, 437 U.S. 584 (1978) (*id.* at 202). The Board concluded that respondents' claimed invention consists of apparatus that is old in the art coupled with "programming which is nonstatutory" under *Benson* and *Flook* (*ibid.*).

The Court of Customs and Patent Appeals reversed. It observed that respondents were claiming "a machine or apparatus" composed of a "combination of *hardware elements*" which fell "literally within the boundaries of §101" (J.A. 18). The court recognized that this "combination" also involves "a portion of the computer's control store microprogrammed in a particular manner" (*id.* at 19). It left standing the Board's determination that the apparatus was old in the art and that only the microprogram was new (*id.* at 18), but it rejected the Board's "distillation of [the] claim down to the information contained in the firmware * * *" (*id.* at 19). The proper analysis, the court asserted, involves determining "whether the claim recites a mathematical algorithm, and, if so, whether it preempts the use of the algorithm" (*ibid.*). Applying this test, it stated that there was no mathematical algorithm here (*id.* at 20). Although certain calculations were made, they did not "transform the invention as a whole into a method of calculation" because, after the invention's task was completed, there was not present

[14]Thus, the claims "recite[d] apparatus as a matter of form only" (J.A. 202).

a "solution of an equation, such as the new alarm limit in *Flook*, or the equivalent pure binary number, as in *Benson*. * * *" (*ibid.*). Indeed, the court saw "no difference" between respondents' invention "and a strictly mechanical adding machine * * *" (*id.* at 18–19).

SUMMARY OF ARGUMENT

The Board of Patent Appeals applied the principles of *Parker* v. *Flook*, 437 U.S. 584 (1978) to the facts of this case and correctly concluded that respondents' claims did not recite statutory subject matter because, aside from an unpatentable computer-implemented algorithm, everything else in the claims was conventional.

Computer programs, which are simply the expression in language intelligible to the computer of the fundamental ideas of the computer user, are outside of the scope of Section 101, whether viewed as translations or as abstract intellectual concepts. Moreover, since these programs simply describe particular examples of the inherent operations of conventional computers, they are not patentable as novel applications of known machines. Although the algorithms this Court considered in *Flook* and in *Gottschalk* v. *Benson*, 409 U.S. 63 (1972), were designed to solve mathematical problems, those cases do not turn on that fact; all computer-implemented algorithms are equally unpatentable. Nor is it significant that the novel computer program at issue here is firmware (a microprogram stored in a control unit), rather than software (a program stored in a computer's main memory unit). Firmware does not convert the conventional computer,

designed to accept firmware of this type, into a new, patentable machine.

As this Court noted in *Benson* and *Flook*, the decision whether, and by what method and to what extent, protection should be afforded to computer programs is for Congress, particularly since the matter is currently under consideration in that body.

ARGUMENT

I. RESPONDENTS' CLAIMS LACK PATENTABLE SUBJECT MATTER

This Court in *Parker* v. *Flook*, 437 U.S. 584 (1978), prescribed a two-step test for determining whether a claimed invention is statutory subject matter under Section 101. First, the claim must be searched to determine whether it recites a scientific principle, idea, concept, formula, or phenomenon of nature (J.A. 591–594). That "basic tool [ ] of scientific and technological work" (*Gottschalk* v. *Benson*, *supra*, 409 U.S. at 67) should be "assumed to be within the prior art" (437 U.S. at 594). Second, what remains must be analyzed to determine whether it is novel (see *O'Reilly* v. *Morse*, 56 U.S. (15 How.) 61, 115 (1853)). If what is left is old, the claim does not recite statutory subject matter. *Parker* v. *Flook*, *supra*, 437 U.S. at 594–595. Applying this test, the Board found that the claims here recited an unpatentable algorithm for implementation by a digital computer, that everything else in the claims was conventional, and therefore that the claims were nonstatutory (J.A. 202).

This was a straightforward, and correct, application of the legal principles of *Parker* v. *Flook* to the facts of this case. The hardware comprising respondents' in-

vention is well known. Indeed, in their amended application respondents referred to prior art[15] to describe the principal hardware components they utilized (J.A. 129–134), including, among others, the main memory (J.A. 132), scratch pad memory (J.A. 132–133), and control store unit (J.A. 133–134). Nor can there any longer be a substantial claim that respondents' arrangement of these components is novel. The examiner and Board of Appeals rejected that claim (J.A. 151, 200), and the court below did not disturb that rejection (J.A. 18). To be sure, respondents claim here (Br. in Opp. 14–15, 19), as they did unsuccessfully in the court below (J.A. 18), that they have created a unique hardware arrangement. They are unable, however, to show that the Patent Office's findings are clearly erroneous.[16] Indeed, respondents admitted in their ap-

---

[15] *E.g.*, prior patent applications, books, and existing machines.

[16] They did not attempt to do so in their opposition to the petition for certiorari, preferring instead simply to assert that "on the record before this Court, the Respondents' invention is acknowledged to be a new machine as claimed" (Br. in Opp. 3 n.7). This is clearly incorrect. Although in his final denial of the application the examiner did not rely on the Hauck patent as prior art (see note 10, *supra*), he did rely on the conclusion that the "environment" of the claim was "a multi programming computer system comprised of * * * well known components" and that "any life which the claim may have in view of that which is admittedly old in the art, resides in the * * * 'switch-system base instruction" (J.A. 151–152). The Board agreed (*id.* at 198–200, 202). Respondents recognized this in their notice of appeal to the court below (*id.* at 205). It was precisely because respondents' claims disclosed no novel arrangement of hardware that their application was rejected by the examiner and the Board. As noted, *supra*, page 7, the court below left the premise of this rejection undisturbed, but found the combination of

plication that other inventors, not they, had earlier obtained a patent whose specification disclosed a hardware arrangement identical to respondents'.[17] Moreover, respondents do not now dispute acknowledging in their application that Claude Carre and others had previously invented a computer machine with the same hardware arrangement now claimed in respondents' diagrams (compare Br. in Opp. 12 n.22 with Pet. 8–9 n.12). Thus, whatever inventiveness inheres in respondents' claimed digital computer must reside either in the algorithm itself or in the algorithm coded in firmware that is itself old in both form and arrangement. In neither case have respondents claimed statutory subject matter.

A. COMPUTER PROGRAMS ARE NOT PATENTABLE.

This Court suggested in *Benson* and more emphati-

---

old hardware, conventionally used, with the microprogram to be patentable (J.A. 18–20).

[17]To overcome an initial rejection on the ground that their description of the invention was not sufficiently specific under 35 U.S.C. 112, respondents, in an amendment, stressed that their arrangement of hardware components had already been patented (J.A. 136). Thus, they observed that in their specification "Figures 1–14h are identical" to those in U.S. Patent No. 3,891,974 assigned to Honeywell, and that "the specification describing these figures is substantially the same" (*ibid.*). (The application for U.S. Patent No. 3,891,974 was filed by Coulter and Migdalek on December 17, 1973, more than a year before Respondents' application). In fact, Figures 1–14i, not "h", are the same in the issued patent and in Respondents' specification. The only difference is in the addition of Figures 15a–c in Respondents' specification to depict the microprogram (J.A. 120; see, also, Pet. 8 n.12 discussing the prior Carre patent application and Pet. App. 52a–53a). A copy of the Coulter patent and of the Carre patent application have been lodged with the Clerk of this Court.

cally indicated in *Flook* that traditional computer programs—*i.e.*, software—are unpatentable. *Benson*, *supra*, 409 U.S. at 72-73; *Flook*, *supra*, 437 U.S at 595. Respondents' assignee Honeywell agrees here, as it did in *Benson*, that computer programs are not and should not be patentable (Br. in Opp. 2). This is in accord with basic patent law principles for several reasons.

First, every algorithm implemented by computer programmers is essentially the expression of a fundamental idea embodied in a flow chart, itself translatable without further creative thought into a set of instructions to the computer.[18] See Breyer, *The Uneasy Case For Copyright: A Study Of Copyright in Books, Photocopies, And Computer Programs*, 84 Harv. L. Rev. 281, 341-342 (1970). Thus, respondents' idea for transferring data between the scratchpad registers and the main memory of a digital computer is embodied in the algorithm represented by the flow charts of Figures 15b and 15c of their specification (J.A. 120) and is easily programmable by persons of ordinary skill in the art (J.A. 138, 155, 169).[19] Neither the fun-

---

[18] Our discussion here, as throughout this brief, is limited to algorithms in the computer programming context. See note 12, *supra*. We do not, therefore, suggest that a "step-by-step procedure for obtaining a given result" (C. Sippl & C. Sippl, *supra*, at 23) necessarily is unpatentable as an algorithm when no computer program is involved. In other contexts, a description of a patentable process could be similarly characterized. See *Flook*, *supra*, 437 U.S. at 589.

[19] Respondents contend that "[f]igures 15b and 15c are flow diagrams of machine operations" and "do *not* illustrate microprogram steps" (Br. in Opp. App. B 2b). Before the Patent Office, however, respondents claimed that the control memory

damental idea nor the translation of that idea into language that can be comprehended by the computer—the computer program—is patentable. The program itself is no more patentable than any other translation from one language to another; the fundamental idea is no more patentable than new ideas for step-by-step methods of doing business, which have repeatedly been held unpatentable. See *Joseph E. Seagram & Sons* v. *Marzall*, 180 F.2d 26, 28, (C.A.D.C. 1950) (method of testing consumer product preferences); *Hotel Security Checking Co.* v. *Lorraine Co.*, 160 F.2d 467, 469 (2d Cir. 1908) (bookkeeping system); *Berardini* v. *Tocci*, 190 F. 329, 332–333 (S.D.N.Y. 1911), aff'd on the opinion below, 200 F. 1021–1022 (2d Cir. 1912) (system using code messages for banking transactions); *Loew's Drive-In Theatres, Inc.* v. *Park-In Theatres, Inc.*, 174 F.2d 547, 552 (1st Cir. 1949) (idea of self-service cafeteria); *In re Patton*, 127 F.2d 324, 327–328 (C.C.P.A. 1942) (fire protection system); *In re Wait*, 73 F.2d 982 (C.C.P.A. 1934) (method of selling stock).

There is an important difference between patenting a novel process and patenting an algorithm for a computer program. With the novel process the inventor seeks to preclude others from actually carrying out the

---

"when microprogrammed in accordance with the microprogram depicted by flow charts of 15b and 15c results in the firmware * * *" (J.A. 141–142). Respondents repeatedly referred to the microprogramming of the control memory by the "microprogram as disclosed by the flow charts of Figures 15b and 15c," noting that "persons of ordinary skill in the art can microprogram them when provided with the flow charts of the microprogram" (*id.* at 138, 155, 169).

procedures of his invention. *Cochrane* v. *Deener*, 94 U.S. 780, 788 (1876). Nothing in the patent law, however, would prevent others from writing down and passing on the description of his process as long as the process itself was not practiced. But with the algorithm for a computer program the inventor seeks to prevent others from even communicating his new idea by translating his algorithm into a program and loading the program into a computer. In other words, he seeks to patent the idea itself, or at least a form of expressing it.[20]

Second, the rationale of *Parker* v. *Flook* is not limited, as the court below suggested, to computer-implemented algorithms designed to solve mathematical problems—*i.e.*, to manipulate data in order to arrive at a solution to a numerical problem (J.A. 17-18). The court's distinction between "mathematical" and "non-mathematical" computer-implemented algorithms is, as the Board observed (J.A. 202), basically illusory: because a digital computer is designed solely to perform mathematical computations, any program that it can implement must necessarily be mathematical.[21] The fact that many programs—like elementary alge-

---

[20] Hence, Congress is presently considering a proposal for clarifying the copyright status of computer programs (see page 24, *infra*).

[21] The CCPA recognized this in *In re Sherwood*, 613 F.2d 809, 816-817 (1980), petition for cert. pending, when it stated that: "writing a computer program may be a task requiring the most sublime * * * inventive faculty or it may require only the droning use of a clerical skill. The difference between the two extremes lies in the creation of mathematical methodology [the algorithm] to bridge the gap between the information one starts with ('the input') and the information that is desired ('the out-

bra—are abstractions of physical phenomena that might also be expressed non-mathematically, in words, does not make such programs any less mathematical. Nor are computer programs "non-mathematical" when the result of the computer's completion of the mathematical calculations directed by the program is to supply the computer user with non-numerical information—such as a page of the Milwaukee telephone directory or the text of a court opinion (J.A. 18).[22]

The computer industry has not generally drawn the distinction between "mathematical" and "non-mathematical" computer-implemented algorithms that the CCPA apparently[23] finds significant.[24] Moreover,

---

put')." But whatever the level of inventiveness, the creation of "mathematical methodology" is not patentable, as *Flook* and *Benson* indicate.

[22] It is, of course, possible for a process utilizing a computer to be patentable; if, after setting aside the computer-implemented algorithm, what remains is new and useful, a patentable claim is asserted. *Flook*, *supra*, 437 U.S. at 591-592.

[23] It is not clear that the Court of Customs and Patent Appeals finds this difference to be of general significance for it has stated that a claim whose only arguable invention was a mathematical algorithm is nonetheless patentable subject matter. *In re Johnson*, 589 F.2d 1070 (C.C.P.A. 1978). It may be more accurate to conclude, as respondents did in their argument to the Board, that subsequent decisions of the Court of Customs and Patent Appeals have restricted this Court's decision in *Benson* "to its own fact situation" (J.A. 175; cf. J.A. 146, 160).

[24] In *Gottschalk* v. *Benson*, the inventors did not limit their definition of an algorithm to mathematical equations. In their application to the Patent Office, Benson and Tabbot defined "algorithm" to mean "a self-consistent set of ordered steps specifying definable operations and leading to a desired result." Record on Appeal, *In re Benson*, 441 F.2d 682 (C.C.P.A. 1971),

nothing in *Benson* or *Flook* requires or suggests a distinction in the patent law between types of computer-implemented algorithms. Both cases involved "mathematical algorithms" as the CCPA has used that term, but both decisions rested on earlier cases finding lack of patentable subject matter in claims for abstract intellectual concepts or scientific truths—some mathematical in nature, some not. *Gottschalk* v. *Benson*, *supra*, 409 U.S. at 63-64; *Parker* v. *Flook*, *supra*, 437 U.S. at 584-585. See *Funk Brothers Seed Co.* v. *Kalo Inoculant Co.*, 333 U.S. 127, 130 (1948); *Rubber-Tip Pencil Co.* v. *Howard*, 87 U.S. (20 Wall.) 498, 507 (1874); *Wyeth* v. *Stone*, 30 F. Cas. 723 (C.C.D. Mass. 1840) (No. 18,107) (Storey, J.); *O'Reilly* v. *Morse*, *supra*, 56 U.S. (15 How.) at 112-113. See, also, *Graham* v. *John Deere Co.*, 383 U.S. 1, 8 n.2 (1966). Both *Benson* and *Flook* instead turned on the recognition that the essence of computer algorithms—whether they be labeled mathematical or non-mathematical—is that they are mathematical statements of ideas or concepts, and these are not patentable. *Benson*, *supra*, 409 U.S. at 67-68.

Finally, a new computer program is not patentable on the theory that it describes a new use for a known machine. This Court has consistently held that a claim

---

at 34. Similarly, in Honeywell's *Benson amicus* brief in this Court, the definition of "algorithm" was not limited to the solution of mathematical problems. That brief stated that an algorithm is a "prescribed set of well-defined rules or processes for the solution of a problem in a finite number of steps, *e.g.*, a full statement of an arithmetic procedure for an evaluation, to a stated degree of precision." Brief of Honeywell, Inc. as Amicus Curiae, *Gottschalk* v. *Benson*, 409 U.S. 63 (1972), at 12 n.18.

describing the inherent operation of a known machine is not statutory subject matter. *Corning* v. *Burden*, 56 U.S. (15 How.) 252, 268 (1853); *Risdon Iron & Locomotive Works* v. *Medart*, 158 U.S. 68, 77, 84 (1895); *Westinghouse* v. *Boyden Power Brake Co.*, 170 U.S. 537, 554–557 (1898); *Expanded Metal Co.* v. *Bradford*, 214 U.S. 366, 383 (1909).[25] But that is exactly what is claimed by respondents and others seeking patents on computer programs—whether the claim is to a process patent or, as here, is to a machine patent. The inventors of the digital computer described in respondents' specification intended that their machine could be programmed in many different ways.[26] Indeed, the capacity of the control store unit to be microprogrammed so as to control various operations of the computer probably contributed to the usefulness of this particular computer hardware arrangement in the first place. And whether the machine is capable of executing respondents' algorithm is obviously predetermined by the design of the machine. Employing a new computer program (whether that program is conventional software or a new mi-

---

[25] The same rule has been applied in the lower courts. See, *e.g.*, *Newton Steel Co.* v. *Surface Combustion Co.*, 75 F.2d 305, 308 (6th Cir. 1935); *Vapor Car Heating Co.* v. *Gold Car Heating & Lighting Co.*, 7 F.2d 284, 287–288 (2d Cir. 1925); *In re Wadman*, 94 F.2d 993, 997–998 (C.C.P.A. 1938).

[26] Respondents themselves acknowledged the versatility of the control store unit when they observed that it can be programmed "for a variety of central processing unit CPU 1306 operational modes" (J.A. 60; see, also, Coulter patent No. 3,891,974, *supra*, note 12, at 23) (using identical language to describe the capability of the control store unit). The examiner concurred (J.A. 182).

croprogram of the kind respondents have claimed) thus does not change how the designer of the computer contemplated it would operate. It merely describes one inherently available operation of the already-patented machine.[27] In these circumstances, a patent on a computer program, or—as respondents claim—on the computer machine with the microprogram, would in effect grant a further patent on an old machine. See *Detroit Gasket & Mfg. Co.* v. *Fitzgerald Mfg. Co.*, 89 F.2d 178, 183 (2d Cir. 1937). This would extend the term of the original patent on the machine, despite the public's right of free access when the statutory patent term expires. *Graham* v. *John Deere Co.*, 383 U.S. 1, 6 (1966). Plainly, then, such claims are for the inherent functioning of a known machine and are, accordingly, not patentable subject matter.[28]

In sum, respondents' idea for transferring information from the scratchpad memories to the main memory, as implemented in their novel microprogram, is not within the scope of patentable subject matter as

---

[27] It is as if respondents claimed a new machine from the combination of an old player piano and a piano roll inscribed with a song they had created.

[28] Even if an application identifying a novel use for an old machine might be patentable as a process patent (see *In re Zierden*, 411 F.2d 1325, 1328–1331 (C.C.P.A. 1969)), respondents have not discovered such a new use. The idea of switching data from scratchpad registers to main memory by means of a computer program is not respondents'. Respondents themselves have noted that model-dependent software previously had been used for this purpose (J.A. 26–27). In any event, such a discovery would not be patentable as a machine. See, *e.g.*, *Roberts* v. *Ryer*, 91 U.S. 150, 157 (1875); *General Electric Co.* v. *Jewel Incandescent Lamp Co.*, 326 U.S. 242, 247–249 (1945).

defined in Section 101 and must, under *Parker* v. *Flook*, be "treated as though it were a familiar part of the prior art" (437 U.S. at 592). Because everything else in their claims is old in the art, the Board correctly applied *Parker* v. *Flook* in concluding that the claims do not recite statutory subject matter.

B. ALGORITHMS CODED IN FIRMWARE ARE NOT PATENTABLE.

The present case differs from *Benson* and *Flook* in that it does not involve a software computer program and traditional process claims. Rather, respondents have embodied their program in firmware and assert that they wish to patent a machine. These distinctions, however, do not entitle respondents to a patent.

The fact that the program is in firmware rather than traditional software is of no legal significance. As the Court of Customs and Patent Appeals itself noted, firmware "may be likened to software" (J.A. 19). And see *In re Maucorps*, 609 F.2d 481 (C.C.P.A. 1979). The only difference between the two is that with software the program is loaded into a different type of memory storage from the type of storage—a control unit—used in firmware (compare J.A. 7-8 with J.A. 9). A program directing a series of sequential steps is the essence of both. The algorithm is no less abstract when claimed as a part of firmware.[29] The control unit,

---

[29] In the instant case it effectuates no physical transformation of material, defines no procedure the computer machine was not already capable of performing, contemplates no post-solution activity, and has no practical application except in connection with a digital computer. See *Gottschalk* v. *Benson*, *supra*, 409 U.S.

like a blank page of a book, is capable of being imprinted with a variety of instructions (see note 7 and page 17, *supra*).[30] It contains the algorithm's directions to the components of the computer just as the page of a cookbook could be printed with directions to a cook for using specialized kitchen equipment to simplify traditional recipes. Like the directions in the cookbook, however, the microprogram in the control unit cannot be patented no matter how original the algorithm upon which it is based. See *In re Russell*, 48

---

at 71–72. Moreover, the algorithm is not limited to the particular digital computer described in respondents' specification. It may be used in other computers as well (see page 5, *supra*). Although respondents now claim (Br. in Opp. 19, n. 34) that "[t]he invention [*i.e.* the firmware] is marketed as an integral part of a new computer machine," they emphasized before the Board that their invention included "a *model-independent* instruction means which is implemented in firmware" (J.A. 165; emphasis added; see also J.A. 27). In sum, what respondents seek to protect is simply their idea for transferring data from one place to another inside a known computer.

[30] Before this Court, however, respondents argue that the PROM described in their specification requires "physical alterations by the manufacturer of the computer to create the desired circuits" (Br. in Opp. App. B 1b). They offer no citation to the record in support of this proposition. On the contrary, the language of the specification (J.A. 60, quoted at note 7, *supra*) as well as respondents' description in their specification of a "control store interface adapter C1A 1302", a "control and load unit CLU 1304", and a "control store loader CSL 1303" from which "the control store unit CSU 1301 receives microinstructions" (J.A. 60; see Fig. 13a, J.A. 112) strongly suggest that the control unit can be programmed electrically by the person using the computer in the field. The record does not clearly indicate which type of PROM is involved here, although the capacity of respondents' PROM to be "reloaded" (J.A. 60) suggests that it is of the reprogrammable type. See R. Kline, *supra*, note 6, at 333. For instance, see U.S. Patent No. 3,944,984 to Morley, *et al.*, patented March 16, 1976, filed April 23, 1974.

F.2d 668, 669 (C.C.P.A. 1931) (printed directory not patentable); *In re Kothny*, 96 F.2d 289, 291 (C.C.P.A. 1938) (novel measuring scale on transparent paper not patentable); *Murphey* v. *American Ry. Express Co.*, 24 F.2d 318, 320 (8th Cir. 1928) (novel means for waybilling an express shipment using a unique label not patentable); *Wier* v. *Coe*, 33 F. Supp. 142, 143–144 (D.D.C. 1940), aff'd per curiam, 122 F.2d 959 (D.C. Cir. 1941) ("musically staffed sheets" and novel method of separately identifying musical themes and their recurrence not patentable).

Nor does respondents' microprogram somehow transform the old hardware into a new machine (cf. Br. in Opp. 19). The fact that the unpatentable algorithm in the microprogram was recorded in the control unit no more operates to make it statutory subject matter than would the recording of an unpatentable process on a blank tape make the recorded process patentable. See *In re Walter*, 205 U.S.P.Q. 397, 409 (C.C.P.A. 1980). Still less does the recording of the microprogram in the control unit turn the combination of the programmed unit and the conventional elements of the computer into a new machine. The only machine involved here is the familiar general purpose digital computer, unchanged by respondents in any of its hardware components or their arrangement. To hold the microprogrammed computer patentable as a new machine would be to make patentability turn on the skill of the draftsman. But *Parker* v. *Flook* teaches that the congressional directive in Section 101 of the Patent Law requires that every patent grant contain a patentable invention. While claimants can draft their

claims as they like, the Patent Office must determine what, if anything, the claimants actually invented. For the purposes of this inquiry, substance must govern form. See *Parker* v. *Flook*, *supra*, 437 U.S. at 590, 594. The Patent Office correctly found that the only thing that respondents invented was a computer algorithm. That, however, is not statutory subject matter. As Judge Rich has elsewhere noted, "the legal doctrine that a new program makes an old general purpose digital computer into a new and different machine * * * partakes of the nature of a legal fiction when it comes to drafting claims." *In re Johnston*, 502 F.2d 765, 773 (C.C.P.A. 1974) (Rich, J. dissenting). That "fiction" should not be permitted to prevail over the requirement for patentable subject matter in Section 101. See Report of the President's Commission on the Patent System 13 (1966), as quoted in *Gottschalk* v. *Benson*, *supra*, 409 U.S. at 72.[31]

II. IT IS FOR CONGRESS TO DETERMINE WHETHER, AND THE EXTENT TO WHICH, COMPUTER PROGRAMS OUGHT TO BE PROTECTED

This Court has twice stressed that if the patent laws are to be extended to cover computer programs this must be done by Congress, not the courts. Noting that the President's Commission on the Patent System rejected a proposal that computer programs be made

---

[31] Similarly, it is immaterial that respondents have chosen to claim a patent on a machine rather than a process, and phrase their claims in machine rather than process terms. The principles of *Parker* v. *Flook* are as applicable to machine claims as to process claims. *In re Freeman*, 573 F.2d 1237, 1247 (C.C.P.A. 1978); *In re Maucorps*, *supra*, 609 F.2d at 485; *In re Walter*, 205 U.S.P.Q. 397, 407–408 (C.C.P.A. 1980). Cf. *Benson*, *supra*, 409 U.S. at 67–68.

patentable, the Court in *Gottschalk* v. *Benson*, *supra*, 409 U.S. at 72–73, concluded that "considerable problems are raised which only committees of Congress can manage * * *."[32] And in *Parker* v. *Flook*, *supra*, 437 U.S. at 595, the Court emphasized the "[d]ifficult questions of policy" Congress could answer "on the basis of current empirical data not equally available to this tribunal." These policy questions include the decision whether the public interest would be served by opening the patent system to an industry that has grown remarkably without such protection, and the relationship of any federal law to foreign laws that do not permit the patenting of computer programs. In addition, Congress is the proper body to address in the first instance the difficulties that would have to be overcome in order to develop an effective system for the examination of computer programs and the determination of when a particular algorithm infringes a protected one.[33]

---

[32] See "To Promote the Progress of . . . Useful Arts," Report of the President's Commission on the Patent System 13 (1966) (reprinted in Subcomm. on Patents, Trademarks, and Copyrights, Senate Comm. on the Judiciary, Doc. No. 5, 90th Cong., 1st Sess. 21 (1967). The ABA Patent, Trademark and Copyright Law Section agreed that there should be "a legislative resolution of many of the uncertainties in the law relating to proprietary rights relating to computer programs, rather than a case by case resolution in the courts." ABA Patent, Trademark and Copyright Law Section, *1972 Committee Reports* 251 (1973) and *1972 Summary of Proceeding* 150 (1973).

[33] These policy questions and problems are discussed in the President's Commission Report, *supra*, note 22, and in briefs amicus curiae filed by the Computer and Business Equipment Manufacturers Ass'n in *Parker* v. *Flook*, *supra* (No. 77-642, 1977 Term, at 17–21, 25–26) and by Honeywell, Inc. in *Gottschalk* v. *Benson*, *supra* (No. 71-485, 1972 Term, at 17–18).

These issues are further complicated by recent developments pertaining to the copyrighting of computer programs.[34] The National Commission on New Technological Uses of Copyrighted Works has recommended amendment of the copyright law to make explicit that computer programs are the proper subject matter of a copyright and that the copyright laws apply to all computer uses of copyrighted programs.[35] On March 26, 1980, Representative Kastenmeier introduced a bill entitled The Computer Software Copyright Act of 1980. 126 Cong. Rec. H2263 (daily ed.). Enactment of that bill would resolve some of the problems addressed in the National Commission report. This current legislative activity demonstrates the wisdom of this Court's determination in *Benson* and *Flook* to leave to Congress the determination of the appropriate method and level of protection, if any, for computer programs.

---

[34]The present copyright statute provides that "[c]opyright protection subsists * * * in original works of authorship fixed in any tangible medium of expression * * * from which they can be perceived, reproduced or otherwise communicated, either directly or with the aid of a machine or device." 17 U.S.C. (Supp. II) 102. The Copyright Office has been registering computer programs since 1964 (see Copyright Office Circular 311 (January 1965)) but their copyright status remains in doubt. See Pope and Pope, *Protection of Proprietary Interests In Computer Software*, 30 Ala. L. Rev. 527, 546 (1979).

[35]See Final Report of the National Commission on New Technological Uses of Copyrighted Works 1-2 (1979).

CONCLUSION

For the foregoing reasons, the judgment of the Court of Customs and Patent Appeals should be reversed.

Respectfully submitted.

WADE H. MCCREE, JR.
Solicitor General

SANFORD M. LITVACK
Assistant Attorney General

LAWRENCE G. WALLACE
Deputy Solicitor General

HARRIET S. SHAPIRO
Assistant to the Solicitor General

ROBERT B. NICHOLSON
FREDERIC FREILICHER
Attorneys

JOSEPH F. NAKAMURA
Solicitor

THOMAS E. LYNCH
Associate Solicitor
United States Patent and Trademark Office

June 1980

☆ U. S. GOVERNMENT PRINTING OFFICE: 1980 321673 7

79-855

Supreme Court, U. S.
FILED
JUL 25 1980
MICHAEL RODAK, JR., CLERK

NO. 79-855

IN THE

Supreme Court of the United States

OCTOBER TERM, 1980

SIDNEY A. DIAMOND,
COMMISSIONER OF PATENTS AND TRADEMARKS,
Petitioner,

V.

JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN,
Respondents.

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

BRIEF FOR THE RESPONDENTS

NICHOLAS PRASINOS
Honeywell Information systems Inc.
Office of the General Counsel
200 Smith Street
Waltham, Massachusetts 02154
(617) 895-6129
Counsel for Respondents

Of Counsel:
FAITH F. DRISCOLL
HENRY L. HANSON
LAWRENCE J. MARHOEFER
RONALD T. REILING

Washington, D.C. • CLB PUBLISHERS' • LAW PRINTING CO. • (202) 393-0625

NO. 79-855

IN THE

Supreme Court of the United States

OCTOBER TERM, 1980

SIDNEY A. DIAMOND,
COMMISSIONER OF PATENTS AND TRADEMARKS,
Petitioner,

V.

JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN,
Respondents.

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

BRIEF FOR THE RESPONDENTS

QUESTIONS PRESENTED

The question presented by the Commissioner is hypothetical. There is no computer program disclosed or claimed by the Respondents. An accurate formulation of the questions presented is as follows:

1. Is a computer machine created by the art and industry of man and which has not always existed a "machine" as that term is used in 35 USC 101?

2. Is a new man made computer machine comprising elements arranged in a unique combination wherein each

element cooperates and functions with each other to mutually achieve an improved result patentable subject matter within the meaning of 35 USC 101?

PARTIES TO THE PROCEEDING

The caption shows all the parties. However the inventors, John J. Bradley and Benjamin S. Franklin (Respondents), have assigned (Reel 3188, Frames 124, 125) the invention to Honeywell Information Systems Inc. by a written instrument dated April 8, 1975 and April 17, 1975 respectively.

TABLE OF CONTENTS

Page

QUESTIONS PRESENTED i

PARTIES TO THE PROCEEDING ii

CONSTITUTIONAL AND STATUTORY PROVISIONS INVOLVED 1

STATEMENT 1

I. WHAT THE INVENTION IS 1

II. WHAT THE INVENTION IS NOT 6

III. A SIMPLIFIED ANALOGY OF THE NATURE OF THE INVENTION 10

IV. PATENT AND TRADEMARK OFFICE CONSIDERATION 13

V. CCPA CONSIDERATION 14

SUMMARY OF ARGUMENT 16

ARGUMENT 20

VI. THE GOVERNMENT SEEKS BY THIS CASE TO PROHIBIT PATENTS FOR COMPUTER RELATED INVENTIONS 20

VII. ENUMERATED SUBJECT MATTER IN 35 USC 101 IS STATUTORY SUBJECT MATTER, PROVIDED IT IS A CREATION OF MAN AND DOES NOT PREEMPT SOMETHING WHICH HAS ALWAYS EXISTED 22

VIII. THIS COURT IN BENSON, FLOOK AND CHAKRABARTY REAFFIRMED THAT INVENTIONS MADE BY MAN ARE PATENTABLE SUBJECT MATTER UNDER 35 USC 101 30

IX. THE COMMISSIONER IS ASKING THIS COURT TO MAKE FINDINGS

ON FACT ISSUES FOR WHICH NO FACTUAL RECORD HAS BEEN DEVELOPED.....37

1. There Is No Finding Of Fact On the Record That Any Program Is Disclosed Or Claimed By The Respondents.....37

2. The Commissioner Seeks To Raise New Fact Issues Before This Court For Which There Are No Findings of Fact on the Record.....38

CONCLUSION.....44

TABLE OF AUTHORITIES

Cases:

Aro Manufacturing Co., Inc., et al v. Convertible Top Replacement Co., Inc.,
365 US 336, 128 USPQ 354 (1961).....12,19,35

Consolidated Electric Light Co. v. McKeesport Light Co.,
159 US 465, 16 S. Ct. 75 (1895).....34

Corning, et al v. Burden,
56 US 252, 15 How. 252, 14 L. Ed. US 683 (1853) ...9

Diamond v. Chakrabarty,
____ US ____ , Opinion No. 79-136 (1980) .....*passim*

Eibel Process Co. v. Minnesota Ontario Paper Co.,
261 US 45, 43 S. Ct. 322 (1923).....26,28

Electric Railroad Signal Co. v. Hall Railway Signal Co.,
114 US 87, 5 S. Ct. 1069, 29 L. Ed. 96 (1885).....12

Funk Brothers Seed Co. v. Kalo Innoculant Co.,
333 US 127, 68 S. Ct. 440 (1948) 28

Gottschalk v. Benson,
409 US 63, 93 S. Ct. 253, 175 USPQ 673 (1972).... *passim*

Graham v. John Deere Co. of Kansas City,
383 US 1, 86 S. Ct. 684 (1966)..... 41

Great Atlantic & Pacific Tea Co. v. Supermarket Equipment Corp.,
340 US 147, 71 S. Ct. 127 (1950) 21

Gustafson, In re,
331 F.2d 905, 141 USPQ 585 (1964) 10

Hazeltine Research, Inc. v. Brenner,
382 US 252, 86 S. Ct. 335, 147 USPQ 429 (1965)... 41

International Curtis Marine Turbine Co. v. William Cramp & Sons Ship & Engine Bldg. Co.,
228 US 645, 33 S. Ct. 722, 202 F. 932 PA, (1913)... 27

Kewanee Oil Co. v. Bicron Corp.,
416 US 470, 94 S. Ct. 1879 (1974) 21

Kropa v. Robie, et al,
187 F.2d 150, 88 USPQ 478 (1951)..... 4

Kruger v. Whitehead,
332 US 774, 68 S. Ct. 105, 153 F.2d 238, C.C.A. Cal. (1946) 27

LeRoy v. Tatham,
14 How. 155 (1852) 27

Lincoln Engineering Co. of Illinois v. Stewart-Warner Corp.,
303 US 545, 58 S. Ct. 662 (1938) 10

Mackay Radio & Telegraph Co. v. Radio Corporation of America,
306 US 86, 59 S. Ct. 427 (1939)..... 27

Mayo Knitting Machine & Needle Co. v. E. Jenckes Mfg. Co.,
133 F. 527 R.I. (1904)..... 27

O'Reilly v. Morse,
56 US 62, 15 How. 62-137, 14 L. Ed. 601 (1853).... 28

Parker v. Flook,
437 US 584, 98 S. Ct. 2522 (1978) ..... *passim*

Reekendorfer v. Faber,
92 US 347, 23 L. Ed. 719 (1875) 10

Rubber-Tip Pencil Co. v. Howard,
20 Wall. 498, 507 (_____) 27

Ryan, et al v. Goodwin, et al,
3 Sumner C.C.R. 514 (1839) 34

Safety Car Heating & Lighting Co., Inc. v. General Electric Co., et al,
155 F.2d 937, 69 USPQ 401 (1946) 36,37

Sears, Roebuck & Co. v. Stiffel Co.,
376 US 225, 84 S. Ct. 784 (1964) 21

The Telephone Cases, 126 US 1, 8 S. Ct. 778 (1888) 28

U.S. v. Adams, et al,
383 US 39, 86 S. Ct. 708 (1966) 34,36,37

Worrest, In re,
201 F.2d 930, 96 USPQ 381 (1953) 10

Statutes:

35 USC 100 25
35 USC 101 ..... *passim*
35 USC 102 ..... *passim*
35 USC 103 ..... *passim*
35 USC 112 9,33,39
1 Stat 109 (1790) 23

Miscellaneous:

Computer Dictionary and Handbook, Charles J. Sippl & Charles P. Sippl, Howard W. Sams & Co. Inc., N.Y. (1972) 2

Deller's Walker on Patents, 2nd Edition (1964) 27,40

Edison, His Life, His Work, His Genius, by William Adams Simonds, Blue Ribbon Rooks, Inc., New York, Copyright 1934 by the Bobbs-Merrill Company 34

Journal of the Patent Office Society, Vol. XVIII, No. 4 (1936) "Operation of the Patent Act of 1790", by P.J. Frederico 28

Journal of the Patent Office Society, Vol. 59, p. 737.....4

Manual of Patent Examining Procedure of the U.S. Patent & Trademark Office 706.03(j).....9

Microcomputer Dictionary and Guide, Charles J. Sippl and David A. Kidd, Matrix Publishers Inc., Champaign, Ill. (1975).....3

Microprogramming: Principles and Practices, Samir S. Husson, Prentice-Hall Inc., Englewood Cliffs, N.J. (1970).....4

Patent Law Fundamentals, P. Rosenberg, (1975).....31

Robinson, The Law of Patents, (1890).....5,12

The Law of Chemical, Metallurgical and Pharmaecutical Patents, Howard I. Forman, Editor, published by Central Book Company N.Y. (1967).....26

Webster's Collegiate Dictionary, Fifth Edition.....26

Webster's New International Dictionary, Second Edition...25

Webster on Patents (1844).....30

CONSTITUTIONAL AND STATUTORY PROVISIONS INVOLVED

Article I, Section 8, Clause 8, of the Constitution of the United States provides:

> The Congress shall have Power ***
> To promote the Progress of Science and useful Arts by securing for limited Times to Authors and Inventors the exclusive right to their respective Writings and Discoveries. . . .

The relevant statute, 35 USC 101, is correctly set forth in the Commissioner's Brief.

STATEMENT

Certiorari was granted as to a patent application for a computer machine which had been rejected by the Board of Appeals of the United States Patent and Trademark Office (hereafter "Patent Office" or "PTO") because of the allegation that the invention was a computer program. The rejection was unanimously reversed by the United States Court of Customs and Patent Appeals (hereafter "the Court below" or "CCPA").

I.

WHAT THE INVENTION IS.

The Respondents' invention is one of a series of inventions which collectively define an entirely new computer machine which is now being commercially marketed worldwide as the Honeywell Series 60, Level 64 computer. This computer

was developed through the joint efforts of technical staffs located in the United States and France. Their efforts extended over several years, involved hundreds of engineers and development costs exceeded $25 million dollars. Respondents' invention is a machine. It comprises register elements, hardware gates, logic circuits and memory elements which are permanently incorporated into the Level 64 computer. The individual hardware elements of the invention cooperate and function with each other to automatically and substantially instantaneously change the physical capabilities of the Level 64 computer.

With the advent of multiprocessing[1] and multiprogramming,[2] the physical characteristics, internal structure and operational capabilities of computers became very complex. One major problem was the reinitialization of such computers.[3] To facilitate computer usage it was necessary to further improve them. By instantaneously changing the computer's internal physical characteristics for accommodating different job requirements, Respondents' invention improves computer efficiency. For example, the physical capabilities of the overall computer that are necessary to solve business problems are different than those required to

---

[1]Multiprocessor — a computer with multiple arithmetic and logic units for simultaneous use and capability. See *computer Dictionary and Handbook* by Charles J. Sippl and Charles P. Sippl, page 285.

[2]Multiprogramming — The concurrent execution of two or more programs by a single computer (JA-87) or, a technique for handling numerous routines or programs simultaneously by overlapping or interleaving their execution; i.e., permitting more than one program to time-share machine components. See *Computer Dictionary and Handbook* by Charles J. Sippl and Charles P. Sippl, page 285.

[3]Reinitialization is necessary when it is desired to change the architecture of a computer for accommodating different jobs; or when the computer system "crashes" i.e. it becomes passive.

solve scientific problems. The invention can rapidly change the architecture of the computer from that of a business type architecture to a scientific type. This is just one of many different improvements that the invention provides. The invention does this by enabling the electronic switches of the system base structure to connect or disconnect various structures of the computer without having to reinitialize the computer system. The invention eliminates the need for complete reinitialization of the overall computer system involving the setting of all the electronic switches within the computer and associated peripheral subsystems required for transforming a passive computer into an active computer — an operation which has always been time-consuming.

The various elements of the Respondents' claimed combination invention are described in the specification[4] and shown in Figures 1-15 of the drawings.[5] One element of the claimed combination is referred to as a system base structure[6] which is typically comprised of integrated flip-flop circuits,[7] or as a series of integrated registers shown in photograph (e) of the Joint Appendix.[8] Another element of

---

[4]JA 24-97.

[5]JA 99-120.

[6]JA-103, Figure 6.

[7]The Microcomputer Dictionary and Guide, by Charles J. Sippl and David A. Kidd on page 129 defines flip-flops as follows:

> "Basic logic gates can be grouped together to form components that are capable of storing and/or altering binary data. The most common component used for storage within the arithmetic unit of the computer is the flip-flop. A flip-flop is a sequential switching circuit capable of storing one bit (binary digit) of information. It has two stable states."

A flip-flop therefore can be likened to a switch that is either on or off.

[8]JA-9.

the claimed combination is a "scratchpad memory" or "scratchpad register".[9] The scratchpad registers (scratchpad memory designated by numeral 1315 of Figure 13a)[10] have been incorporated in the central processing unit (CPU) which is a different structure from that of the system base structure located in the main memory module designated by numeral 102 of Figure 1.[11] Still another element of the hardware combination comprising the invention is a microprogrammed hardware control element[12] shown on Figures 13a and 13b of the Respondents' application,[13] which is commonly referred to as firmware.[14] The microprogrammed

[9]The phrases "scratchpad registers" and "scratchpad memory" are included in the preamble portion of the Respondents' combination claims and is also recited in the body of such claims so as to give life, meaning and vitality to the other elements recited therein. For a complete discussion of limitations as aplied to claim preambles, see *Kropa v. Robie,* et al, 187 F.2d 150, 88 USPQ 478 (1951).

[10]JA-112.

[11]JA-99.

[12]As stated on pages 19 and 20 in the authoritative book on microprogramming entitled, "Microprogramming: Principles and Practices", by Samir S. Husson, Prentice-Hall Inc.:

> "Microprogramming was proposed as an orderly technique for designing and organizing control signals which represent the states of the computer and the orderly transition between these finite states. This was a direct replacement of the classical sequential logic circuits which were designed to interpret a fixed data hardware environment. This view places microprogramming within the domain of the system designer."

[13]JA 112-113.

[14]Respondents' patent application defines "firmware" as that part of a hardware unit implementation which utilizes a form of microprogrammed control (JA-84) and defines "microprogram" as a set of machine codes used to implement the control function of a processor (JA-87). The authoritative sources relied upon by the Commissioner also recognize that firmware is a machine part. In the Journal of the Patent Office Society, Vol. 59, p. 737 it is stated that:

> "A generalized example of firmware is 'a small, easily replaceable part which is pre-punched or wired in such a way that when activated... cause[s] a particular sequence of computer operations to take place. . . .'"

hardware control element also includes hardware structure which is physically located within different parts of the computer.[15] Another claimed element, a hardware gate HG (reference numerals 1507 and 1514)[16] is disclosed in Figure 15b.[17]

The hardware control element in conjunction with the other elements of the invention enables the computer to perform the series of machine operations illustrated in the flow charts of Figures 14a through 14i and Figures 15b and 15c.[18] (The hardware control element may be likened to the cam of an internal combustion engine in that it is essential in carrying out the proper sequence of operations of the machine — it carries out the rule of action of the machine).[19] Other descriptions and illustrations of the Respondents' hardware combination are contained in the application, particularly Figures 5, 8 and 12.[20]

---

[15]"Each of the functional units within the computer includes decoding logic circuits for generating the requisite subcommands (control signals) specified by a control store word under the control of the system clock source." (JA 67-68). Additional structure interconnects the functional units and the microprogrammed control element and the decoding logic circuits and the control element JA-112, JA-113.

[16]JA-120.

[17]*Ibid.*

[18]JA 114-120. Flow charts are a shorthand method for disclosing machines as well as programs. Some typical machine patents utilizing flow charts for disclosure are as follows: U.S. Patent Nos. 3,329,937 (1967); 3,736,566 (1973); 4,183,084 (1980); 4,093,990 (1978) and 3,735,864 (1973).

[19]See *Robinson, The Law of Patents,* Sections 173 and 175 (1890).

[20]JA-102, 104, 111.

II.

WHAT THE INVENTION IS NOT.

The Commissioner has not understood the invention at any point in the history of this case.[21] He has alleged at various times that the invention is a program,[22] a mathematical algorithm and a method of calculation[23] and it is an old combination.[24] In his Brief For The Petitioner, he reiterates his basic theme that computer related inventions are mathematical.[25]

The cornerstone of his argument is that all digital computer machines are mathematical in nature and therefore nonstatutory subject matter under 35 USC 101. The Commissioner has stated in the record below:

> "Since digital computers normally operate in some numeral radix, binary, binary coded decimal, or the like, we consider the operation of Appellants' claimed invention to be mathematical."[26]

Aside from his bare allegations and misinterpretations of terms used by those skilled in the art utilized to disclose the invention,[27] the Commissioner has not identified where in

---

[21]Refer to the plethora of erroneous statements by the Commissioner identified in Respondents' Appendix. R1a - R4a.

[22]PW 34a, 38a, 39a (The letters PW followed by a number indicate the page number in the Commissioner's Petition for aWrit of Certiorari to the United States Court of Customs and Patent Appeals).

[23]PW 29a.

[24]PW 5, 6, 26a, 27a.

[25]The Commissioner states: "[b]ecause a digital computer is designed solely to perform mathematical computations, any program that it can implement must necessarily be mathematical", p. 14.

[26]JA-202.

[27]Refer to Respondents' Appendix for the identification of some of the Commissioner's misinterpretations.

Respondents' application a computer program, a formula, method of calculation, a mathematical algorithm, a law of nature, an idea or a scientific principle is disclosed or claimed.[28] Hence, there is no basis in the record for his conclusion.

The Respondents' claims do not involve a mathematical algorithm; they do not involve any so-called "computer-implemented algorithm"—a phrase coined by the Commissioner. As a matter of fact, the Respondents' claims do not involve an algorithm or procedure of any kind.

One of the errors evident throughout the Commissioner's Brief is his misunderstanding and misuse of the term "algorithm". This Court has defined algorithm as "[a] procedure for solving a given type of mathematical problem."[29] The Commissioner, in coining the term "computer-implemented algorithm" attempts to encompass steps involved in both the creation of a machine and a mathematical algorithm.[30] This is absolutely wrong—both legally and logically.

A mathematical algorithm which brings to light for the first time a phenomenon which has always existed, e.g. Newton's formulation o the law of universal gravitation $F=mm/d^2$ is not patentable subject matter.[31] The algorithm merely

---

[28]The board said that the claims do not *directly* recite a mathematical formula, algorithm, or method of calculation, but, nevertheless, held the claims to be mathematical in nature. As appears from the quoted portion of the board's opinion, the board regarded the fact that digital computers operate in some number radix as conclusive on the issue of whether the appealed claims recite a mathematical algorithm in the *Benson* and *Flook* sense. The board did not, however, direct attention to any specific formula it thought is utilized, or to what, if anything, the mathematical calculations alleged to be present in the claims are directed. JA-17.

[29]*Parker v. Flook*, 437 US at 585 note 1 (quoting *Benson*, 409 US at 65).

[30]See Brief for Petitioner, p. 15.

[31]*Parker v. Flook*, 437 US 584, 98 S. Ct. 2522, f.n. 15 (1978). "The underlying notion is that a scientific principle, such as that expressed by Respondent's [*Flook's*] algorithm, reveals a relationship that has always existed."

expresses the discovery, e.g. a law of nature. It does not describe a machine nor a portion thereof. The discovery is inseparable from the algorithm which brings to light what has always existed. The discovery therefore is the algorithm.

The Commissioner *cannot,* merely be coining a new term, extend this Court's definition of an algorithm to apply to something created by the art and industry of man which has *not* always existed. The act of man in creating a machine is separate and district from the act of expressing a law of nature by an algorithm. In a machine comprising a combination of elements, the elements are normally described by means of blueprints, flow charts and written explanations. If a nonmathematical algorithm were to be utilized in the disclosure of any of the elements of the machine, it could be likened to the blueprint or flow charts. However, when a patent is obtained for the overall machine combination, it does not preempt the use of any of the individual elements or the blueprints utilized in describing th. They are separable from the creation of the machine combination. Similarly no algorithm utilized in disclosing any element of the machine would be preempted and the public is free to use such algorithm. It too is also separate from the machine. The algorithm is not the machine.

It should further be noted that Respondents' invention is not a computer program of any kind. Computer programs are normally written by computer programmers to be executed by the computer to solve some user problem. They are typically embodied in punch cards[32] or magnetic tape.[33] Computer programs are generally sold or licensed separately and apart from the computer hardware, and circulate in commerce as independent products. The Commissioner, if he felt a program was disclosed and claimed, had ample opportunity in the proceedings below to resolve this factual question. On the contrary, he has admitted that the invention is a machine.[34]

[32]See Figure (a) JA-4.

[33]See Figure (b) JA-5.

[34]JA-3.

Nor is the claimed invention an old combination.[35] In support of this allegation the Commissioner is belatedly raising new factual issues clearly recognized as requiring resolution under 35 USC 102 or 35 USC 103 by citing the Carre et al application and the Coulter Patent No. 3,891,974 in his brief. These fact issues are not properly before this Court. Moreover the Respondents' claimed invention is patentably distinct over the newly cited references; it discloses and includes additional hardware not disclosed or suggested in the cited references.[36]

The Respondents' claims express the elements of their machine combination as means for performing a specified function and such claim language must be construed to cover the corresponding structure described in Respondents' specification and equivalents thereof as 35 USC 112 requires.

The Commissioner cannot ignore the Respondents' claim language and simply treat Respondents' invention as a process for the purpose of ignoring the interrelationship of the claimed machine elements upon which patentability of machine inventions is normally predicated.[37] The record

---

[35]In the Manual of Patent Examining Procedures of the United States Patent and Trademark Office it is stated that:

> "The rejection on the ground of old combination (synonymous with 'exhausted combination') requires the citation of references. . . ."

[36]See pages 37-42, infra, for a detailed discussion of these factual issues.

[37]In *corning v. Bruden,* 56 US 252 at 257, 267 and 269 (1853), this Court articulated the fundamental difference between a process and a machine. It stated that when the result or effect was produced by chemical action or by the operation or application of some element or power of nature or of one substance to another, such modes, methods, or operations were called processes — not a machine. The Court further stated that a machine is a concrete or tangible thing consisting of parts or of certain devices and a combination of devices — a machine is not a principle or idea. Accordingly, when an inventor's specification describes a machine, it would be improper to construe the invention as being for a mere function, effect or result instead of being for a machine which is an authorized subject of a patent.

below shows that Respondents do not disclose or claim a result or the inherent function of a machine or an abstract idea as the Commissioner now alleges but rather a machine for producing a new or improved result.

Furthermore, the claimed invention is not an aggregation of elements such as an eraser being attached to one end of a pencil[38] where the individual parts do not cooperate to produce a new or improved result, i.e. improve the writing or the erasing. The Respondents' combination consists of various elements which cooperate with each other to improve the efficiency of a computer.[39]

III.

A SIMPLIFIED ANALOGY OF THE NATURE OF THE INVENTION.

The nature of the invention may be better understood with reference to an internal combustion engine analogy. Consider a hypothetical invention which when incorporated into an existing automobile engine improves the efficiency of that engine and provides a cleaner exhaust. It achieves this by injecting a preheated mixture of air and fuel to each cylinder of the automobile engine at the appropriate power stroke.

---

[38] *Reekendofer v. Faber*, 92 US 347 (1875).

[39] In *In re Worrest*, 201 F.2d 930, 96 USPQ 381 at 381 (1953) it is stated:

> "Patentable combination may be present where there is correlation, cooperation or coordination resulting in mutuality of achievement of common purpose."

See also *Lincoln Engineering Co. of Illinois v. Stewart Warner Corporation*, 303 US 545, 58 S. Ct. 662 at 664 last column, last 7 lines, (1938) and *In re Gustafson*, 331 F.2d 905, 141 USPQ 585 (1964).

The hypothetical invention utilizes a new hardware combination claimed as comprising the following elements: (a) a centrally located preheating chamber; (b) an internal heating device in the chamber for preheating the air and fuel mixture; (c) metal tubes connecting the chamber to each cylinder of the existing automobile engine; (d) inlet and outlet valves in the preheating chamber for admitting and removing the air and fuel mixture; and (e) a control cam for actuating the valves.

The machine operations that this additional hardware combination performs in cooperation with the existing automobile engine are as follows: (a) opening an inlet valve to admit the fuel into the preheating chamber; (b) opening another inlet valve to admit air into the preheat chamber; (c) closing each of the inlet valves at appropriate times; (d) preheating the fuel air mixture to some specified temperature; and (e) opening the outlet valves in the connecting tubes so that each cylinder receives the preheated mixture at the appropriate time. It is clear that all of these operations are *machine operations* of this new claimed hardware combination. They are not machine operations of the existing automobile engine, and they do not comprise a computer program. These machine operations of this new claimed hardware combination are controlled by the cam — which is a hardware element and part of the new combination of hardware elements of the hypothetical combination invention.

The claimed combination comprises the (a) preheat chamber, (b) preheat apparatus, (c) connecting tubes, (d) valves and (e) cam. All of these elements are required to be attached to an existing automobile engine to provide greater mileage and cleaner burning fuel. The fact that a cam controls the machine operations of this new hypothetical

combination does not make the cam a program,[40] nor does it make the cam the sole inventive entity of the new claimed combination.

In evaluating this hypothetical invention under 35 USC 101, it must be considered as a whole. The arrangement of the elements and how they cooperate and function with each other to produce an improved result must be considered. It is the claimed combination that is the invention and not any one element included therein.[41] In *Flook* the Court reiterated its long standing directive that a "patent claim must be considered as a whole."[42]

---

[40]"Every machine is characterized by its rule of action which resides within the machine itself." *Robinson, The Law of Patents,* Paragraphs 173 and 175 (1890).

The rule of action or mode of operation in this hyupothetical invention (supra) is embodied in the cam. The cam is part of the machine and must be considered as part of the combination. Without all of the elements described the cam performs no useful function. In this sense the mode of operation can be said, speaking loosely in the language of computer usage, to be "programmed" into the machine via the cam. It can similarly be said that the mode of operation of knitting machines, steam engines, metal working machines and endless other machines are "programmed" into the machine and comprise the rule of action of the machine. This does not make the machine a program as the Commissioner alleges. Neither does it make the cam a program.

Justice Matthers in speaking for the Supreme Court explained in *Electric Railroad Signal Co. v. Hall Railway Signal Co.,* 114 US 87, 55 S.Ct. 1069, 29 L. Ed. 96 (1858), (a case dealing inpart with mode of operation of a machine) that:

> "It thus clearly appears that the difference in this particular between the inventionclaimed . . . is a difference in the arrangement of the parts and in the principle of communication. . . ."

So it is with the Respondents' invention.

[41]See *Aro Manufacturing Co., Inc. et al v. Convertible Top Replacement Co.,* 365 US 336, 128 USPQ 354 at 355 (1961).

[42]*Parker v. Flook,* 437 US 584, 98 S. Ct. 2522 at 2528, (1978).

The Respondents' invention is analogous in that it comprises a unique combination of hardware elements that coact and function with each other to improve the efficiency and speed of a computer of which it is a part. One element of the claimed combination, a microprogrammed firmware control element, controls the machine operations of this new combination. This firmware control element is analogous to the cam of the hypothetical claimed combination invention. It is abundantly clear from the Respondents' specification that the firmware control element *alone* cannot be used with any existing computer as alleged by the Commissioner.[43] It is necessary that the whole claimed combination of elements including the various hardware registers, hardware structures and firmware control element be incorporated in a computer.

IV.

PATENT AND TRADEMARK OFFICE CONSIDERATION.

In the PTO the Respondents conclusively proved, and the PTO agreed, that their invention clearly distinguishes over the cited prior art — that it is new under 35 USC 102.[44] Respondents also conclusively proved, and the PTO agreed that their invention was described in full, clear, concise and exact terms as to enable any person skilled in the art to which it most closely pertains, to make and use the same.[45]

---

[43]Brief for the Petitioner, p. 5.

[44]The Examiner's rejection under 35 USC 102 (JA-124) was removed in the Office Action of October 27, 1976 (JA-150-152) in view of Respondents' arguments. (JA-126-152).

[45]The Examiner's rejection under 35 USC 112 (JA-122-123) was removed in the Office Action of October 27, 1976 (JA-150-152) in view of Respondents' arguments (JA-135-143).

The Examiner finally rejected the Respondents' claim under 35 USC 101 as drawn to nonstatutory subject matter based on a misinterpretation of *Gottschalk v. Benson* and a misunderstanding of the invention. Without identifying a program, the Examiner merely stated that the:

> ". . . claims . . . set forth an invention *related* to the processing of data by a programmed computer." (Emphasis added).[46]

The Board of Appeals recognized that the invention as recited in Claim 1 consists of hardware structure in the form of registers or portions thereof and firmware.[47] The Board, however, ignoring the combination as a whole and the specific arrangement of the hardware elements claimed and how they cooperate with each other, has characterized each individual element except the hardware control unit (firmware) as old. It dissected the claimed combination invention down to only one element which it erroneously labeled a mathematical algorithm.[48] The Board, with its broad brush, characterized every digital computer machine as being mathematical and therefore nonstatutory under 35 USC 101, a most obvious misstatement of fact and law.

V.

CCPA CONSIDERATION

The Court of Customs and Patent Appeals in a unanimous decision reversed the Board of Appeals. The Court below

---

[46]JA-123.

[47]JA-198-199.

[48]JA 202.

rejected the reasoning of the Board on the basis that such reasoning leads to the conclusion that any computer related invention must be regarded as mathematical in nature, a conclusion which is not compelled by either *Benson* or *Flook.*[49] No difference was seen between Respondents' claimed invention and a strictly mechanical adding machine which is certainly statutory if claimed in a manner which does not embrace any particular calculation that the machine is capable of making.[50]

As concerns the firmware, the CCPA stated:

> "If Appellants were claiming the information embodied in the firmware or the firmware itself, per se, a different case would be presented. We express no opinion on the statutory nature of such an invention, a question not before us. Appellants are claiming a combination of hardware elements, one of which happens to be a portion of the computer's control store microprogrammed in a particular manner. It is this subject matter with which we must deal."

Pursuant to this Court's reasoning in *Flook,* the Court below looked to the specification to determine the nature of the invention. After careful analysis the CCPA concluded the following:[51]

> "We disapprove the board's distillation of Appellants' claim down to the information contained in the firmware in order to hold it nonstatutory. The invention is not claimed in that manner, and, in this case, we see no reason to view the claim format as a subterfuge for masking the presence of an essentially nonstatutory invention."

---

[49]JA-17.

[50]*Ibid* at 18-19.

[51]JA-19.

SUMMARY OF ARGUMENT

This case presents a narrow question of statutory interpretation of 35 USC 101.

The Respondents applied for a patent for a unique computer machine. Throughout the prosecution of this case the Respondents have insisted that their invention is a machine. The Court below agreed with them.[52] Even the Commissioner agreed that the invention is a machine.[53] It can hardly be denied now that the Respondents' claimed invention is a machine. The question then becomes, is it a machine within the meaning of 35 USC 101 — i.e. is it a machine created by the art and industry of man which has not always existed.[54]

In construing 35 USC 101 this Court has recently observed that:

> "Congress plainly contemplated that the patent laws would be given wide scope" [and that] "The relevant legislative history also supports a broad construction . . . . The Committee Reports accompanying the 1952 Act inform us that Congress intended statutory subject matter to 'include anything under the sun that is made by man.' "[55]

The Respondents' invention is man-made. It has not always existed, and the Commissioner has not shown otherwise. The Court below noted the Commissioner's admission that Respondents' claims do not recite a mathematical formula, algorithm, or method of calculation.[56]

---

[52]JA-18.

[53]JA-3.

[54]*Parker v. Flook*, 437 US 584, 98 S. Ct. 2522 at 2527, footnote 15, (1978).

[55]*Diamond v. Chakrabarty*, ____ US ____, No. 79-136 (1980), at p. 5.

[56]JA-17.

Furthermore, the Commissioner has failed to show that a scientific principle, law of nature or abstract idea is disclosed.

The Commissioner has obviously misunderstood and misconstrued the Respondents' invention. This is illustrated by the more substantive errors made by the Commissioner in his Brief For The Petitioner, some of which are identified and corrected in the Respondents' Appendix.[57]

The broad public policy issues raised by the Commissioner regarding the patentability of computer programs are properly matters for Congress and need not be considered by the Court in this case.

In both *Gottschalk v. Benson,* 409 US 63 and *Parker v. Flook,* 437 US 584, this Court's attention was riveted on a phenomenon which always existed; a phenomenon which was discovered by the inventors, expressed as a mathematical algorithm, and merely brought to light by them. In *Benson* this Court correctly held "the discovery of a novel and useful mathematical formula may not be patented." In *Flook,* this Court correctly held that "an improved method of calculation, even when tied to a specific end use, is unpatentable subject matter under Section 101." The *Flook* invention like a law of nature was not "the kind of discovery that the statute was enacted to protect", that is, it represented a "relationship that has always existed".[58] Conversely, any process, machine, manufacture or composition of matter which is "made by man" must fall within one of the categories of general statutory subject matter of 35 USC 101.

The Respondents' invention plainly qualifies as patenta-

---

[57]See pages R1a - R4a, *infra.*

[58]*Parker v. Flook,* 437 US 584, 98 S. Ct. 2522 at 2527 (1978).

ble subject matter under Section 101 based on the decisions of this Court. It is a nonnaturally occurring machine; a product of human ingenuity. It is part of the unique structure of a completely new computer system designed in response to the needs of the competitive marketplace for a more efficient computer.

In spite of the vast and compelling differences between the Respondents' claimed machine invention and the mathematical formulae of the process claims addressed in *Flook* and *Benson,* the Commissioner still attempts to force Bradley's invention into a nonstatutory procrustean bed. He misapplies *Flook* and *Benson* to reach his erroneous conclusion. In analyzing the machine type of invention for which patent protection is sought, he disregards, contrary to the holdings of this Court in *U.S. v. Adams,* 383 US 39 (1966) and in *Parker v. Flook,* 437 US 584 (1978)[59] the features relied upon for patentability for such statutory class of inventions such as the arrangement of the elements claimed and how each element coacts and functions with other elements of the claimed combination to produce a new or improved result. He dissects the Respondents' claimed combination of elements by alleging that each individual element is old and then argues that a nonstatutory idea remains. Utilizing this erroneous rationale, any invention ever invented can be held to be nonstatutory under Section 101. For example, Edison's incandescent lamp can be

[59]While the Commissioner now argues that Respondents' hardware arrangement was disclosed in an earlier obtained patent (see discussion *infra,* pps. 38-42), this argument masks the Commissioner's basic approach of focusing in on a specific element of a machine combination claim regarded as the novel feature and distilling it down to the idea involved.

reduced to its incandescent principle. One need merely hold that a glass bulb was individually old; the carbonified filaments were individually old; and the vacuum technology was old. Then using the Commissioner's erroneous rationale, the only remaining feature of the incandescent lamp is the principle of incandescence which is a natural law. Hence, Edison's epic invention would have been held nonstatutory by the reasoning of the Commissioner.

This Court has flatly rejected such illogical reasoning. In *Aro Manufacturing Co., Inc., et al v. Convertible Top Replacement Co., Inc.,* a case including the infringement of a combination invention, the Court stated:

> "Combination patent covers only the totality of the elements in the claim; no element, separately viewed, is within the grant; there is no legally recognizable or protected 'essential' element, 'gist' or 'heart' of the invention in a combination patent; no element, not itself separately patented, that constitutes one element of the combination patent is entitled to patent monopoly however essential it may be to patented combination and no matter how costly or difficult replacement may be."[60]

The Respondents' invention is a combination of hardware elements and must be viewed in the totality of its elements. No single element of the combination is sought to be patented.

To bypass his statutory duty of examining the Respondents' invention on its merits, the Commissioner is now apparently attempting to import Section 102 and 103 requirements (novelty and nonobviousness) into Section 101. He is also attempting to import a prior invention question into Section 101. No record below has been

---

[60] *Aro Manufacturing Co., Inc., et al v. Convertible Top Replacement Co., Inc.*, 365 US 336, 128 USPQ 354 at 355 (1961).

established for enabling a determination of these questions. This case does not involve these other conditions and requirements of the patent laws.

ARGUMENT

VI.

THE GOVERNMENT SEEKS BY THIS CASE TO PROHIBIT PATENTS FOR COMPUTER RELATED INVENTIONS.

The Commissioner's insistence on a narrow construction of 35 USC 101 to exclude computer machine inventions is contrary to the Constitutional mandate, the intent of Congress and the decisions of this Court.[61] Without any supporting factual evidence, the Commissioner has now alleged the Respondents' invention is a computer program. This erroneous conclusion is based on a serious misunderstanding of the invention.[62] The Commissioner's unsound position in this case can only be viewed as

---

[61]*Diamond v. Chakrabarty* ___ US ___ (1980), Opinion No. 79-136 at 11, 12. "The subject matter provisions of the patent law have been cast in broad terms to fulfill the constitutional and statutory goal of promoting 'the Progress of Science and the Useful Arts' with all that means for the social and economic benefits envisioned by Jefferson. . . .

Nothing in *Flook* is to the contrary . . . *Flook* did not announce a new principle that inventions in areas not contemplated by Congress when the patent laws were enacted are unpatentable per se.

To read that concept into *Flook* would frustrate the purpose of the patent law. . . ."

[62]Some of the numerous substantive errors of the Commissioner are identified and corrected in the Respondents' Appendix R1a - R4a.

another step in his repeated attempts to obtain a broad holding from this Court that computer related inventions are not patentable. This attempt by the Commissioner to obtain a restrictive interpretation of Section 101 to solve his administrative difficulties in the Patent and Trademark Office[63] cannot be condoned.

This Court has recognized that the patent laws enacted by Congress do promote productive effort which, in turn, improves the economy, increases employment and makes better lives for U.S. citizens.[64] The Court has also observed that the patent system has three objectives: (a) to foster and reward inventions; (b) to promote the disclosure of inventions; and (c) to assure that ideas (inventions) already in the public domain remain there for free use.[65] Thus the patent system functions by adding to the sum of useful knowledge.[66]

The Respondents have advanced the state of the computer art with their invention. They have completely disclosed their invention in the specification for the benefit of the public. They do not seek to remove anything from the public domain. Moreover, the grant of a patent would not remove from the public anything it had freely enjoyed prior to the grant of such patent. The type of invention created by the Respondents — computer machines — has traditionally been considered statutory subject matter.[67] The restrictive

---

[63]See Commissioner's Petition for a Writ of Certiorari to the United States Court of Customs and Patent Appeals, p. 13.

[64]*Kewanee Oil Company v. Bicron Corp.*, 416 US 470, 480 (1974).

[65]*Sears, Roebuck & Company v. Stiffel Company*, 376 US 225, 228-231 (1964).

[66]*Great Atlantic & Pacific Tea Company v. Supermarket Equipment Corp.*, 340 US 147, 152 (1950).

[67]See patents classified in PTO Classes 340.-172.5 and 364.

interpretation of Section 101 espoused by the Commissioner would, if followed by this Court, effectively invalidate all of these existing U.S. patents and prohibit issuance of future patents on computer machines. Congress has expressly renounced such a narrow construction of Section 101.[68]

VII.

ENUMERATED SUBJECT MATTER IN 35 USC 101 IS STATUTORY SUBJECT MATTER, PROVIDED IT IS A CREATION OF MAN AND DOES NOT PREEMPT SOMETHING WHICH HAS ALWAYS EXISTED.

A review of the history and basic concepts of the Patent Act since 1790 which eventually culminated in the 1952 Patent Act, makes it clear that 35 USC 101 is concerned with statutory subject matter, and not with a question of novelty. For subject matter to be statutory under Section 101, it must not only fall in one of the categories enumerated but it must be a creation of man and must not preempt something which has always existed. It is only in this sense that the word "new" has been imported into Section 101.[69]

[68]*Diamond v. Chakrabarty,* ____ US ____ (1980), Opinion No. 79-136 at 5.

[69]*Diamond v. Chakrabarty,* ____ US ____ (1980), Opinion No. 79-136, at 6. "Thus, a new mineral discovered in the earth or a new plant found in the wild is not patentable subject matter. Likewise, Einstein could not patent his celebrated law that $E=mc^2$; nor could Newton have patented the law of gravity. Such discoveries are 'manifestations of . . . nature, free to all men and reserved exclusively to none.' " *Funk, supra,* at 130.

On the other hand, 35 USC 102 pertains to the novelty of subject matter found to be statutory under Section 101. A statutory invention is required to be further evaluated as to novelty and unobviousness over prior art inventions of others under 35 USC 102 and 103.

The First Congress, sitting in its second session, exercised its delegated power by enacting the Patent Act of 1790.[70] Section 1 of the Act, the precursor of 35 USC 101, provided for patents to be issued to:

> "Any person or persons . . . [that] hath or have invented or discovered any useful art, manufacture, engine, machine, or device, or any improvement therein not before known or used. . . ."

This section of the Act was further refined 3 years later by the Patent Act of 1793. The word "new" was included and the phrase "not known or used before the application" was added in the paragraph which provided for patents to be granted to:

> "Any person or persons . . . [that] have invented any new and useful art, machine, manufacture, or composition of matter, or any new and useful improvement on any art, machine, manufacture, or composition of matter not known or used before the application. . . ."

The pertinent part of Section 6 of the 1836 Patent Act which parallels Section 1 of the 1793 Act was enacted as follows:

> "And be it further enacted, that any person or persons having discovered or invented any new and useful improvement on any art, machine, manufacture, or composition of matter, not known or used by others before his or their discovery or invention thereof, and not at the time fo application for a patent, in public use

---

[70] 1 Stat 109, April 10, 1790.

> or on sale, with his consent or allowance, as the inventor or discoverer; and shall desire to obtain an exclusive property therein, may make application in writing to the Commissioner of Patent, expressing such desire and the Commissioner, on due proceedings had, may grant a patent therefor."

This Act included language similar to the 1793 Act — but sets forth conditions and requirements relating to novelty— "not known or used 'by others' before his or their discovery or invention. . .".

The 1952 Patent Act in essence separated Section 6 of the 1836 Act into two separate and distinct sections, one relating solely to statutory subject matter and the other relating to novelty.[71]

---

[71]The content of Section 101 of the 1952 Act was included in Section 4886 of the Revised Statutes (35 USC 31, 1946 ed.) which stemmed from Section 24 of the Patent Act of 1870. The language in section 6 of the Patent Act of 1836 was practically the same as that of 1870 in most respects, although it was still in terms of authority to grant a patent rather thaen the right to obtain a patent. The language similar to Section 6 of the 1836 Act was split into two sections and included in Title 35 as follows, Section 101 relating to the subject matter for which patents may be obtained:

> "Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title."

and Section 102 defining statutory novelty and stating other conditions for patentability.

> "A person shall be entitled to a patent unless — (a) the invention was known or used by others in this country, or patented or described in a printed publication in this or a foreign country, before the invention thereof by the applicant for a patent . . . or (b) the invention was described i.. a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent. . . ."

It is illuminating to note that despite innumerable revisions and amendments to the Patent Acts of 1790, 1793, and 1836, that the original language pertaining to categories of patentable subject matter remains remarkably similar.[72]

Although all of the Patent Acts refer to invention or discovery, or both, it does not follow that every invention or discovery is patentable subject matter under 35 USC 101.[73] No discovery of something that has always existed, but was not known until brought to light by the discoverer will entitle him to a patent under 35 USC 101.[74] Laws of nature, physical phenomenon and abstract ideas, though they may be discovered, have been held not patentable because they have always existed—they are not new.[75] It is only in this sense, that something has not always existed, that "new" has been imported into or is still residuum in Section 101.[76] something has not always existed, that "new" has been

---

[72]However the following definitions were added: 35 USC 100(a) defines "invention" to mean invention or discovery while 35 USC 100(b) defines "process" to mean process, art or method and to include a new use of a known process, machine, composition of matter or material.

[73]Webster's New International Dictionary, Second Edition, unabridged at p. 1305 defines Act of Inventing as "[d]evising or contriving up that which has not before existed, as the invention of gunpowder", whereas on page 745 it defines Discover as follows: "To obtain for the first time sight or knowledge of, as of a thing existing already, but not perceived or known."

[74]"An example of such a discovery [of a scientific principle] was Newton's formulation of the law of universal gravitation, relating the force of attraction between two bodies, F, to their masses, m and m', and the square of the distance, d, between their centers, according to the equation $F\text{-}mm'/d^2$. But this relationship always existed — even before Newton announced his celebrated law. such 'mere' recognition of a theretofore existing phenomenon or relationship carries with it no rights to exclude others from its enjoyment." *Parker v. Flook*, 437 USC 584, 98 S. Ct. 2522 (1978).

[75]*Parker v. Flook*, 437 US 584, 98 S. Ct. 2522 at 2528 (1978).

imported into or is still residuum in Section 101.[76] Conversely, "even though a phenomenon of nature or mathematical formula may be well known, an inventive application of it by man may be patentable".[77] This Court has repeatedly held that the method or means of application of a principle or law of nature producing a useful result is the proper subject for a patent.[78] Deller expresses this concept as follows:

> "It is only when the discoverer has gone beyond the mere domain of discovery, and has laid hold of the new principle, force or law and connected it with some particular medium or mechanical contrivance by which, or through which, it acts upon the material world, that he can secure the exclusive control of it through the Patent Act. He then controls his discovery through the means by which he has brought it into practical action, or their equivalent, and only through them. It is an invention, although it embraces a discovery . . . . Discovery can be made of a law of nature, for instance, but it can never be invented. When discovered, a law of nature may be utilized in an invention relating to

---

[76]As stated by Mr. Federico, the architect of the 1952 Patent Act,

> "The question of novelty to others is not particularly relevant at this point as persons may, and often do, invent things which have been made before by others."

"Section 101: Subject Matter for Patents", by P.J. Federico in The Law of Chemical, Metallurgical and Pharmaecutical Patents, Howard I. Forman, Editor, published by Central Book Company, New York (1967).

Note also that Webster's Collegiate Dictionary, Fifth edition defines "new" as "having existed, or having been made, but a short time."

Thus the conditions and requirements of novelty are only relevant under 35 USC 102.

[77]*Parker v. Flook*, 437 US 584, 98 S. Ct. 2522 (1978).

[78]*Eibel Process Co. v. Minnesota & Ontario Paper Co.*, 261 US 45 (1923).

> patentable subjects such as processes, machines, manufactures, or compositions of matter. It is the invention of one or more of these subjects, for the purpose of utilizing a law of nature, and not the discovery of that law, that may be rewarded with a patent."[79]

There is a long-standing recognition of patentable subject matter if the inventor "connects a discovery with some particular medium or mechanical contrivance."

Similar treatment has been accorded to ideas. While a mere idea is not patentable, the means for carrying out an idea is patentable. In such a case the invention consists of the conception of the idea and the means for putting it in practice and producing the desired result.[80] Respondents note that the Commissioner has articulated this same view on the patentability of ideas.[81]

---

[79]Deller's Walker on Patents, Second Edition, Vol. 1, p. 102.

[80]*Mayo Knitting Machine & Needle Co. v. E. Jenckes Mfg. Co.*, 133 F. 527, R.I. (1904); *Kruger v. Whitehead*, 153 F.2d 238, C.C.A. Cal. 91946), *cert. denied*, 332 US 774, 58 S. Ct.; *International Curtis Marine Turbine Co. v. William Cramp & Sons Ship & Engine Bldg. Co.*, 202 F. 932, Pa. (1913) reversed on other grounds, 228 US 6545, 33 S. Ct. 722

[81]"Since the first patent law was enacted in 1790, the patent system has been directed solely to the pragmatic application of ideas. Thus, patents have been issued only with respect to specific and tangible machinery, articles of manufacture, chemical and other physical compositions and processes for making or using such machinery, articles of manufacture and compositions in a specific way. These four classes alone — 'machines', 'articles of manufacture', 'compositions' and 'processes' for making or using them, 35 USC 101—are the categories of statutory subject matter. Principles and ideas, as contrasted with the specific and tangible application of those ideas to a specific use, have always constituted nonstatutory subject matter which cannot qualify for the monopoly grant embodied in a patent, *Leroy v. Tatham*, 14 How. 156, 174-175; *rubber-Tip Pencil Co. v. Howard*, 20 Wall. 498, 507; *Mackay Radio & Tel. Co. v. Radio Corporation*, 306 US 86, 94." Brief for the Petitioner, in *Gottschalk v. Benson*, pp. 17-18.

This view is in keeping with Jefferson's concepts relating to ideas who, as Secretary of State at the time of the enactment of the Patent Acts of 1790 and 1793, was one of the "Patent Board" members[82] who profoundly influenced the fundamental course of the Patent Laws.

The concept articulated by Jefferson that an application of a principle or the reduction to practice of an idea can be patented has been continually affirmed by this Court in such landmark cases as: *O'Reilly v. Morse,* 56 US 62 (1853); *The Telephone Cases,* 126 US 1, (1888); *Eibel Process Co. v. Minnesota & Ontario Paper Co.,* 161 US ____ (1923) *Funk Bros. Seed Co. v. Kalo Co.,* 333 US 127 (1948); *Gottschalk v. Benson,* 409 US 63 (1972); *Parker v. Flook,* 437 US 584 (1978); and *Diamond v. Chakrabarty* ____ US __ (1980) Opinion No. 79-136.

In these cases the Court's attention was clearly focused on phenomena which is required as having always existed, i.e. ideas, laws of nature, scientific principles, or mathematical formulae. In deciding *Flook,* this Court correctly perceived that the mathematical algorithm involved was "a procedure for solving a given type of mathematical problem" and therefore a discovery of something which always existed and

---

[82] "The power to grant patents under the Patent Act of 1790 was completely vested in a board or commission consisting of three members; The Secretary of State, the Secretary of War, and the Attorney General . . . . Thomas Jefferson was also an inventor of considerable merit, having invented a number of ingenious devices. . . . Jefferson became an enthusiastic supporter of the patent system, but before his intimate contact with its beginnings, *he was opposed to monopolies of all kinds* . . . . Subsequent experience, including his personal contact with patents, convinced him of the importance of th patent system, and that an inventor should be rewarded by a limited monopoly . . . ." (Operation of the Patent Act of 1790, by P.J. Federico, Journal of the Patent Office Society, April, 1936, Vol. XVIII, No. 4.

in that sense "a familiar part of the prior art". That is, it disclosed and claimed only the discovery of a formula which like a law of nature is not the kind of discovery that the patent statutes were enacted to encompass. The relationship codified into mathematics by *Flook* always existed but was not known until he brought it to light.[83]

The Respondents' invention in this case is not a mathematical formula but a machine. Stated differently, it is not for a principle (idea) but for a machine which carries a principle (idea) into effect.[84] The invention has not always existed; it

[83] *Parker v. Flook*, 437 US 584, 98 S. Ct. 2522 and 2527 and footnote 15 (1978).

> "The rule that the discovery of a law of nature cannot be patented rests, not on the notion that natural phenomena are not processes, but rather on the more fundamental understanding that they are not the kind of "discoveries" that the statute was enacted to protect. . . .
>
> "The underlying notion is that a scientific principle, such as that expressed in Respondents' algorithm, reveals a relationship that *has always existed.*"

[84] The above statement is consistent with the reasoning of the English case of *Neilson, et al v. Hasford*, Webster's Patent Cases 273 (1844) relied upon by this Court in the *Morse* case and in *Flook*. In *Neilson* the Court reviewed the specification of a patent entitled "An invention for the improved application of air to produce heat in fires, forces and furnaces where bellows or other blowing apparatus are required". One objection raised by the defendants charged with infringement of the patent was that the specification did not sufficiently ascertain and describe the nature of the invention. *Plaintiffs' counsel argued:* It has been suggested that this is a patent merely for a principle; but that is not so; it is a patent for the mode of carrying a principle into effect. The mode of heating air and increasing combustion was known before; this patent is taken out for the novel application of air so heated to certain useful purposes - for passing the air in a heated state, instead of a cold state as formerly, into furnaces; and the mode of operation is by interposing a closed vessel, exposed to heat, between the blowing apparatus and the furnace. That is a sufficiently practical discovery to be the subject of a patent, according to all the authorities: *Minter v. Wells* (1 C.M. &

(continued)

was not known in the Section 101 sense — it is new. Accordingly, it is patentable subject matter under Section 101.

VIII.

THIS COURT IN *BENSON, FLOOK* AND *CHAKRABARTY* REAFFIRMED THAT INVENTIONS MADE BY MAN ARE PATENTABLE SUBJECT MATTER UNDER 35 USC 101

In interpreting the subject matter that is patentable under Section 101, this Court noted that Congress intended that the patent law should be given wide scope.[85] This intent is set forth in the Committee Reports accompanying the 1952 Patent Act and in the testimony of P.J. Federico, a principal draftsman of the 1952 Act. Specifically, "The Committee Reports . . . inform us that Congress intended statutory

---

(footnote continued from preceding page)

R505), *Hornblower v. Boulton* (8 T.R. 98), Webster on Patents, p. 136.

The English Court concluded:

Then, taking the construction of this specification on ourselves, as we are bound to do, it becomes necessary to examine what the nature of the invention is which the plaintiff has disclosed by this instrument. It is very difficult to distinguish it from the specification of a patent for a principle, and this at first created in the minds of some of the Court much difficulty; but, after full consideration, we think that the plaintiff does not merely claim a principle, but a machine embodying a principle, and a very valuable one. We think the case must be considered as if, the principle being well known, the plaintiff had first invented a mode of applying it by a mechanical apparatus to furnaces; and his invention then consists in this - the interposing a receptacle for heated air between the blowing apparatus and the furnace.

[85] *Diamond v. Chakrabarty* ____ US ____, Opinion No. 79-136 at 5.

subject matter to 'include anything under the sun that is made by man.' This same language was employed by P.J. Federico, . . . in his testimony . . . 'Under Section 101 a person may have invented a machine or manufacture, which may include anything under the sun that is made by man. . . .'[86] The Court also reiterated that laws of nature, physical phenomenon and abstract ideas are not patentable, citing *Flook* and *Benson* among other decisions.

It should be noted that the inventions before the Court in *Benson* and *Flook* were mathematical formulae.[87] In determining that *Flook's* invention was not the kind of discovery that the statute was enacted to protect, the Court noted that the "underlying notion is that a scientific principle such as that expressed in Respondent's [Flook's] algorithm, reveals a relationship that has always existed."[88] This is consistent with the proposition that in granting patent rights the public must not be deprived of any rights that it previously enjoyed.[89] In *Benson*, the Court correctly noted that the process claimed was an attempt to patent an abstract idea and that the public's right to use the mathematical formula had to be preserved.[90]

---

[86]*Diamond v. Chakrabarty* ____ US ____, Opinion No. 79-136 at 5. References to legislative history have been deleted from the quoted language.

[87]"The only novel feature of the method is a mathematical formula. In *Gottschalk v. Benson*, 409 US 63, 175 USPQ 673, we held that the discovery of a novel and useful mathematical formula may not be patented. The question in this case is whether the identification of a limited category of useful, though conventional, post-solution applications of such a formula makes Respondent's method eligible for patent protection." *Parker v. Flook*, 437 US 584, 98 S. Ct. 2522 at 2527 (1978).

[88]*Ibid* at note 15, Brackets added.

[89]P. Rosenberg, Patent Law Fundamentals, Section 4 at 13 (1975).

[90]*Gottschalk v. Benson*, 409 US 63 (1972).

The Respondents' case, as did *Flook Benson* and *Chakrabarty* turns entirely on the proper construction of Section 101 which sets forth the subject matter that is eligible for patenting. Section 101 does not involve the issues of novelty and obviousness that arise under Sections 102 and 103. This court in *Flook* made it clear that in interpreting 35 USC 101, the "obligation to determine what type of discovery is sought to be patented must precede the determination of whether that discovery is in fact new or obvious."[91]

In construing Section 101 in *Flook* this Court carefully reviewed the patent application and concluded:

> "The patent application does not purport to explain how to select the appropriate margin of safety, the weighting factor, or any of the other variables. Nor does it purport to contain any disclosure relating to the chemical processes at work, the monitoring of process variables, or the means of setting off an alarm or adjusting an alarm system. All that it provides is a formula for computing an updated alarm limit."[92]

Based upon this evaluation, the Court concluded that *Flook's* process invention was unpatentable subject matter under 35 USC 101 because the invention considered as a whole only pertains to subject matter that has always existed. The formula disclosed was assumed not to be new; it was treated as a familiar part of the prior art. Since the Respondents in *Flook* "had not challenged the examiner's finding that, the formula is the only novel feature of Respondents' method"[93] their application treated as a whole contained no patentable subject matter under 35 USC 101.

---

[91]*Parker v. Flook*, 437 US 584, 98 S. Ct. 2522 at 2527 (1978).

[92]*Ibid* at 2524.

[93]*Parker v. Flook*, 437 US 584, 98 S. Ct. 2522 at 2525.

Respondents submit that the record below establishes that the Commissioner has admitted that Respondents disclose and claim a "machine" invention and not a program.[94] The record also indicates that the Respondents have disclosed the hardware elements of their invention in sufficient detail to satisfy the requirements of 35 USC 112.[95] It is obvious that the Respondents' invention was created by the art and industry of man. Before they conceived and implemented their invention, it did not exist. When they invented it they advanced the state of the art and disclosed it in their specification thereby adding to the sum of useful knowledge. Thus there is no benefit that the public previously enjoyed and which would be taken away by the granting of a patent for the Respondents' invention.

Because complex technological advances (e.g. micro-programmed control elements) make it difficult for unskilled persons to identify differences in machine structure, such difficlties should not be used as the basis for concluding that an invention not fall within one of the statutory classes of 35 USC 101. Historically, this Court has always recognized that even the slightest structural change can form the basis of patentability from both a prior art point of view as well as from an inventorship point of view. As Justice Bradley commented when he placed Edison in his rightful place as the inventor of the incandescent lamp:

> "It seems to us that they (Sawyer and Man) were following a wrong principle — the principle of small resistance in an incandescing conductor and a strong current of electricity; and that the great discovery in the

---

[94]JA-3.

[95]See Joint Appendix pages 151-152, where the Examiner withdrew his rejection of claims under 35 USC 112.

art was that of adapting high resistance in the conductor with a small diminution in the strength of the current. This was accomplished by Edison in his filamental thread-like conductors. This was really the grand discovery in the art of electric lighting without which it could not have become a practical art for purposes of general use in houses and cities."[96]

In 1895, this Court affirmed the decision of the lower Court in the McKeesport case.[97]

Concomitantly, it can be said that Fleming added only one small wire to Edison's incandescent lamp to create his vacuum tube detector, known as the "Fleming valve". The Fleming invention was followed by the efforts of deForest, who added merely a third wire midway between the two elements of "Fleming's valve". These small, barely discernible changes in structure gave birth to patentable inventions which revolutionized the electronics industry and greatly benefited mankind.

Historically, this Court in regard to assessing allegations that an invention claimed is not new in the case of combination inventions has required examination of the invention of the combination as a whole rather than any one ingredient or element.[98] Further, when there were allegations

---

[96]*Edison*, His Life, His Work, His Genius, by William Adams Simonds, Blue Ribbon Books Inc., New York, Copyright 1934 by the Bobbs-Merrill Company.

[97]In *Consolidated Electric Light Co. v. McKeesport Light Co.*, 159 US 465, 16 S. Ct. 75 (1895), claims of the Sawyer-Man Patent No. 317,076 covering the use of all fibrous and textile materials for incandescent conductors were held to be void.

[98]In the case of *Ryan, et al v. Goodwin, et al*, 3 Sumner C.C.R. 514 (1839) involving an infringement of a patent for a new and useful improvement in the manufacture of friction matches for the instantaneous production of light in the point of defense that the

(continued)

that the invention claimed involved a scientific principle or law of nature this Court employed a similar examination to ensure that claims were not directed to something that nature created, but rather that the invention was for a human accomplishment or creation, the construction of something which did not always exist before.

Contrary to this Court's holding in *Flook*, the Commissioner has ignored key hardware elements of Respondents' unique combination claims not disclosed in any references cited by him.[99] He further characterizes each individual element as old. He then alleges that Respondents' apparatus is old in the art[100] and ignores the arrangement and cooperation of the individual hardware elements with each other in mutually achieving an improved result.[101]

To illustrate the error of Commissioner's reasoning by analogy, assume that one is considering the following old elements: (a) wheels, (b) shafts, (c) rope, and (d) a frame.

A first combination which can be claimed is a cart that is

(footnote continued from preceding page)

invention claimed is not new stated: As to the first point, it is mainly question of fact. It is certainly not necessary, that every ingredient, or, indeed that any one ingredient used by the patentee in his invention, should be new or unused before the purpose of making matches. The true question is whether the combination of materials by the pantee is substantially new. Each of these ingredients may have been used for matches, or combined with other materials for other purposes. But if they have never been combined together in the manner stated in the patent, but the combination is new, then, I take it, the invention of the combination is patentable.

Additionally, see *U.S. v. Adams, et al*, 383 US 39, 86 S. Ct. 708 (1966) and *Aro Manufacturing Co., Inc., et al, v. Convertible top Replacement Co., Inc.*, 365 US 336, 128 USPQ 354 (1961).

[99]See discussion pps. 40, 41, *infra*.

[100]JA 182-183.

[101]JA-200.

constructed with the wheels, shaft and frame to be pulled along by the rope.

A second claimable combination is a system of pulleys constructed with the wheels, shaft, frame and rope to lift large weights.

A third combination is a steering mechanism constructed with the wheels, shaft, frame and rope to steer a boat.

All of these combinations have the same well known elements. Each element performs no new function. The wheels turn. The shaft supports the wheels. The frame supports the shaft. The rope transmits force. However, each of these elements cooperate with each other to produce a new or improved result. What gives uniqueness to each combination is the arrangement of these known elements with respect to each other and how they cooperate and function with each other, ie., how they are brought together.[102] The proper approach to analyzing combination inventions was eloquently stated by Judge Learned Hand as follows:

> "Substantially all patentable inventions are for combinations of old elements and what counts in determining whether a patentable invention has been made is the selection, out of all their possible permutations, of that new combination which will be serviceable."[103]

If the claimed arrangement of elements and how they cooperate with each other can be ignored by the Commissioner and no prior art need be cited, it is clear that he could, with the same defective logic, hold as nonstatutory any and all combination inventions.

---

[102]This Court in *U.S. v. Adams*, 383 US 39 (1966) said:

> "It begs the question . . . to state merely that magnesium and cuprous chloride were individual known battery elements. If such combination is novel, the issue is whether bringing them together as taught by Adams was obvious in the light of prior art.

[103]*Safety Car Heating & Lighting Co., Inc. v. General Electric Co., et al*, 155 F.2d 937 (1946).

IX.

THE COMMISSIONER IS ASKING THIS COURT TO MAKE FINDINGS ON FACT ISSUES FOR WHICH NO FACTUAL RECORD HAS BEEN DEVELOPED.

1. There Is No Finding Of Fact On The Record That Any Program Is Disclosed or Claimed By Respondents

The Commissioner has grounded his arguments on a false premise — that Respondents' machine is a program. He has educed no facts to support this premise other than his bare allegations and convoluted reasoning. To attempt to convince this Court that the machine is a program he has disregarded *U.S. v. Adams*[104] and *Safety Car Heating & Lighting Co., Inc. v. General Electric Co., et al*[105] and dissects the invention to one element — the firmware control element — and considers it independent of all of the other claimed elements.[106]

He erroneously alleges that the firmware is a "computer

---

[104]See page 36 *supra*, footnote 102.

[105]See page 36 *supra*.

[106]When the type of Respondents' invention claimed has been implemented totally in hardwired circuits, it has been regarded as statutory subject matter under 35 USC 101. The following U.S. Patents are examples of hardwired implementations of similar types of inventions classified in the same technical classes 340, 172.5 and 364 as the Respondents' inventions: 3,105,143; 3,454,933; 3,454,931; 3,425,039; 3,323,110; 3,522,589; 3,344,404; and 3,564,504. Additionally, the following U.S. Patents involve similar types of inventions implemented in "firmware": 3,400,371 and 4,028,670. There are numerous patents which have issued to date on Respondents' type of invention since the development of computer technology.

program that regulates the internal operation of the computer," but he has not shown where in the Respondents' application any computer program is disclosed or claimed. He has not even defined a computer program.

Utilizing the internal combustion engine analogy supra on page 10 of this Brief and applying the Commissioner's erroneous reasoning to it, the combination of hardware can be reduced to the hardware "cam" as the sole novel feature. The cam is then erroneously characterized as a program embodying merely an idea and accordingly is unpatentable. Thus, a hardware element utilized as one element of a combination can be magically transformed by the Commissioner's convoluted reasoning into a program which has never been disclosed or claimed.

2. The Commissioner Seeks To Raise New Fact Issues Before This Court For Which There Are No Findings Of Fact On The Record

In a belated attempt to show that the invention is not novel, he for the first time during the entire prosecution introduces the related patent application of Claude Carre et al Serial No. 528,951 and U.S. Patent No. 3,891,974 to Brent W Coulter, et al, both identified by the Respondents in their application.[107] This importation of novelty regarding the inventions of others into 35 USC 101 can only be viewed as a further and alternate attempt to distill the invention to some alleged unpatentable part, principle or idea.[108]

---

[107]Coulter's patent incorporates by reference the Carre, et al application and issued as a patent subsequent to the filing of Respondents' application. Hence any disclosure therein was not known to the public at the time of filing of Respondents' patent application.

[108]See discussion of alternate modes of Commissioner's distillation of the invention on pages 18, 19 *supra*.

The Respondents incorporated by reference the application of Claude Carre into their application to satisfy the enabling and best mode requirements of 35 USC 112. Both Claude Carre and Coulter worked along with Bradley and Franklin and others in developing the Series 60, Level 64 computer. Both Claude Carre, et al and Coulter et al created inventions which are separate and distinct from Respondents' invention. The patent application covering Carre's invention was subsequently abandoned.

Introducing Carre's abandoned patent application and the Coulter patent for the first time in the Commissioner's brief is improper under Rule 21.1a of the Rules of the Court.[109] Furthermore the resolution of these new fact questions raised by the Commissioner of novelty and prior invention cannot be made by the mere introduction of these documents and without access to the facts and circumstances attendant to these documents. Moreover, such questions are not relevant to the 35 USC 101 issue before this Court. Newness in the 35 USC 101 sense requires only that the subject matter which falls within one of the enumerated categories did not always exist. That is, it is not a law of nature, abstract idea or mathematical formula.[110] Clearly the Respondents' invention is a contrivance of man which did not always exist.

---

[109]It must be noted that the Carre application and the Coulter patent which were both cited to the PTO by the Respondents, were before the PTO during the entire prosecution of Respondents' application. However, neither the examiner nor the Board of Appeals ever raised either the Carre application or Coulter patent as a possible ground of rejection of Respondents' claims under 35 USC 101, 102 or 103.

[110]See *Parker v. Flook*, 437 US 584, 98 S. Ct. 2522 at 2527 and footnote 15 (1978).

The separate question of whether or not an invention is patentable over a body of technical knowledge publicly known is a fact question which requires a comparison of Respondents' invention with such prior art inventions of others under 35 USC 102 and 103. It is not a question within the purview of 35 USC 101.

By citing the new references, the Commissioner is now attempting to raise new grounds of rejection which properly fall under 35 USC 102 and 103 for which no findings of fact have been made on the record below and are not before this Court.[111] In effect, the Commissioner is asking this Court to perform the functions of the PTO and to re-examine Respondents' patent application in view of two nerly cited references and without the benefit of any findings of fact regarding these issues. Respondents have never admitted nor has it ever been established in the record below that either the Carre application or the Coulter patent disclose the combination of hardware elements now being claimed by Respondents.[112] They do not. It should be noted that the

---

[111]The Examiner first rejected Respondents' claims under 35 USC 102 as unpatentable over the Hauck patent. In making this rejection, the Examiner stated: "The apparatus in Figure 1 of Hauck cooperates to perform the function of Applicant's claimed invention." However, when presented with this rejection, similar in substance to the Petitioner's new allegations, Respondents had the opportunity to present persuasive arguments which the Petitioner noted resulted in the Examiner's removal of the rejection.

[112]In general, a mere application for a patent is not a part of the prior art since an application is confidential and the public cannot be informed of its contents until a patent issues upon it. But an application which eventuates into a patent, while such patent may not be part of the prior art as it may have issued after the patent in suit, can nevertheless be used to prove that the patentee of the patent in suit was not the original and first inventor or discoverer of any material and substantial part of the thing patented. (Deller's Walker on Patents, Section 107, page 124).

(continued)

Respondents' disclosure includes Fig. 15a, 15b and 15c,[113] not shown in either Carre or Coulter. For example, Fig. 15b[114] further discloses additional hardware such as a hardware gate HG, reference numerals 1507 and 1514. This hardware gate and how it is used and cooperates with other elements of the claimed combination is further described in Respondents' specification.[115] Additionally Figs. 15b and 15c disclose additional hardware structure and machine operations.[116] This structure is claimed in claims 4-6.[117] Moreover, since Respondents' patent application was filed before the Coulter, et al patent issued, even the common portion of Respondents' invention cannot be considered as a "known machine" or a "familiar general purpose digital computer" as the Commissioner now alleges.

(footnote continued from preceding page)

This Court has held that earlier filed copending patents are "prior art" as that term is used in 35 USC 103 against copending applications of different inventors. However, this finding was based upon the factual record established within the PTO and that there was no dispute that Applicant's invention as well as his application, were made after the cited copending application was filed (*Hazeltine Research, Inc. v. Brenner,* 382 US 252, 86 S. Ct. 335, 147 USPQ 429 (1965).

Consistent with this Court's decision in *Graham v. John Deere Co.,* 383 US 1, the scope and content of the prior art must be determined. The essence of the evidentiary question is what the disclosure of a prior art patent means to one of ordinary skill in the art. Where a portion of that prior art disclosure refers to the state of the art as it existed at the time that the application for the prior art patent was filed, that disclosure likewise must be tested against a standard of what it means to one of ordinary skill in the art.

[113]JA-120.

[114]JA-120.

[115]JA-78, 79.

[116]JA-120.

[117]JA-207

This additional hardware structure and other structure not disclosed in either Carre or Coulter must be given its full weight in considering the invention as a whole. By overlooking such structures and also the microprogrammed control element and how these and other elements cooperate with each other as the Commissioner is now urging, he could hold any invention unpatentable. For example, such revolutionary inventions as Lee de Forrest's electron triode amplifier tube could be held unpatentable by citing Fleming's invention which is identical in structure and arrangement to de Forrest's invention except for a small piece of wire inserted btween the anode and cathode of Fleming's diode. Thus the PTO could ignore the fact that this little piece of wire allowed the other elements to cooperate and function with each other in a manner so different that they would launch mankind into the electronic era. However, this is an improper forum to evaluate novelty in view of the patents of others, since the issue of novelty and unobviousness is not before this Court and entail fact issues which cannot be resolved without the introduction of additional evidence.

Furthermore, the Commissioner in citing the Coulter Patent No. 3,891,974 to show "the same hardware arrangements now claimed in Respondents' diagrams" is tacitly admitting that the Respondents' invention is statutory subject matter. By equating the Respondents' invention to the patented Coulter invention, he implie[illegible] admits that the Respondents' invention also comprises [illegible]utory subject matter which is the sole issue before this Court.[118]

[118]Utilizing Socratic logic the truth of this statement is evident. When the Commissioner compares the Respondents' invention to the Coulter invention and concludes that Coulter ". . . and others had previously invented a computer machine with the same hardware arrangement now claimed in Respondents' diagrams . . ." he acknowledges that the subject matter of both inventions is the same. However, since the Coulter

(continued)

This Court need not decide any broad policy issues with respect to the patentability of computer programs as advocated by the Commissioner. There is no computer program disclosed or claimed by the Respondents. Moreover, as this Court has clearly articulated, such policy issues are to be addressed by Congress.

The Respondents' invention is a creation of man — a new computer machine. It is new in the 35 USC 101 sense in that it has not always existed. It clearly comprises that type of subject matter that the statute was enacted by Congress to protect, as interpreted by the decisions of this Court including *Benson, Flook* and *Chakrabarty.* The invention is a new combination of elements which has not always existed and which elements cooperate and function to achieve an improved result. Such new combinations have always been patentable subject matter under 35 USC 101. The granting of a patent on such improvement combination does not extend the life of any existing patents on the computer it improves. Moreover, merely because such combination is a permanent part of a computer machine does not convert it into unpatentable subject matter under 35 USC 101. Holding

---

(footnote continued from preceding page)

invention has matured into a patent, it must be statutory under 35 USC 101. If the subject matter of the Respondents' invention is the same as the Coulter invention it too must be statutory.

If the Commissioner wishes to rely on the Coulter patent, he should seek a remand in order to apply those references against the Respondents' invention under 35 USC 102 or 103 instead of seeking to import 35 USC 102 or 35 USC 103 issues into the statutory interpretation of 35 USC 101. It is this statutory duty of the Commissioner that he has consistently refused to perform with respect to computer related inventions and insists on melding 102 and 103 requirements in 101.

this invention as statutory subject matter will not deprive the public of any rights that it has heretofore freely enjoyed.

CONCLUSION

Accordingly, it is hereby requested that the decision of the court below be affirmed.

Respectfully submitted,

NICHOLAS PRASINOS
Honeywell Information Systems Inc.
200 Smith Street
Waltham, Massachusetts 02154
(617) 895-6129

Counsel for Respondents

Of Counsel:
FAITH F. DRISCOLL
HENRY L. HANSON
LAWRENCE J. MARHOEFER
RONALD T. REILING

RESPONDENTS' APPENDIX

The Commissioner has presented throughout his Brief erroneous conclusions, misstatements of fact, quotations out of context, and other clearly misleading statements. Respondents have identified below some of the more flagrant errors contained in the Commissioner's Brief and have also presented the correct information. Because of space limitations, no attempt has been made to identify and correct all of the Commissioner's errors.

1. Erroneous Statement of Commissioner: (PB 18)

"A patent on a computer program . . . or on the machine with the microprogramming would in effect grant further patent on an old machine. . . . This would extend the term of the original patent on the machine."

Correct Statement:

This statement discloses a fundamental misunderstanding of the invention and the Patent Laws. The Respondents have created an apparatus which when built into a computer improves computer performance. They are seeking patent coverage only on the claimed improvement as sanctioned by 35 USC 101. Therefore, the notion that a second patent on a separate and distinct invention which is physically incorporated into an earlier patented machine extends the term of the first patent on the machine is false. Consistent with 101, the public is free to use the original machine in any manner that they originally enjoyed prior to the patenting of Respondents' invention.

2. Erroneous Statement of Fact by the Commissioner: (PB 20 Footnote 30)

". . .the language of the specification. . . strongly suggests that the control unit can be programmed electrically by the person using the computer in the field."

Correct Statement of Fact:

The control store unit disclosed in Respondents' specification is a 9K bipolar integrated circuit programmable read-only memory (PROM) mixed with read/write random access store (RAM) (JA 61). Bipolar integrated circuit PROMs are "programmed" by physically "breaking" connections of circuit elements in series with such bipolar transistor circuits. The term "bipolar" does not encompass MOSFETS which are described in the Kline text cited by the Commissioner as being used in reprogramming a PROM (See Microprogramming: Principles and Practices, by Samir S. Husson, p. 153).

Moreover, the "Microcomputer Dictionary and Guide", by Charles J. Sippl and David A. Kidd, Matrix Publishers Inc., Champaign, Il., 1975, has this to say about PROM programming steps. There are generally five necessary steps to program PROMs so that the microcomputer can control various systems.

"1. The designer lists the sequence of mechanical operations needed."

"2. From this sequence of operations a program sheet is written."

Although the "state of the art terms" refer to "programming" or "microprogramming" a ROM or PROM, it is clear that a *designer* utilizes this technique to build "mechanical operations" into the machine.

3. Erroneous Statement of Commissioner: (PB 4, 5)

"An instruction in the firmware developed by Respondents permits the user to communicate with the scratchpad registers and to switch data back and forth from the registers to the system base in the main memory."

Correct Statement of Fact:

The so-called switch base instruction is hardware incorporated into Respondents' computer. It is not included within user programs. Moreover, no data is transferred from the scratchpad memory registers to the system base.

4. Erroneous Statement of Fact: (PB 2, 3)

"The computer's hardware includes a main memory, which has a system base with which the user of a computer can communicate."

Correct Statement of Fact:

The user cannot access the system base. It is only accessible via the boundary address register (BAR) which can be read but not written. (JA53-54). The (BAR) separates an area in main memory reserved for hardware in the system base (JA 54). Customer user programmers do not need a system base structure to communicate with the computer since they have always and can continue to communicate directly with a computer via their own programs.

5. Misleading Quotation of Commissioner: (PB 3, 4)

"Firmware consists of a sequence of basic instructions as a microprogram in the computer's memory."

The above is the Commissioner's quotation from p. 754 of Ross taken from 59 Journal Patent Office Society. This quotation fosters the misconception that firmware is nothing more than a series of instructions.

Additional Quotation from the Commissioner's Identical Source: On p. 737 of 59 Journal Patent Office Society, Mr. Ross makes it unequivocally clear that firmware is a machine part. He states:

"A generalized example of firmware is 'a small, easily

replaceable part which is pre-punched or wired in such a way that when activated . . . cause[s] a particular sequence of computer operations to take place. . . .' "

Other errors of fact were also identified in Respondents' Brief in Opposition of a Writ of Certiorari to the United States Court of Customs and Patent Appeals pages 1b-3b.

Supreme Court, U. S.
FILED
OCT 3 1980
MICHAEL RODAK, JR., CLERK

No. 79-855

In the Supreme Court of the United States

OCTOBER TERM, 1979

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS, PETITIONER

v.

JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

REPLY BRIEF FOR THE PETITIONER

WADE H. McCREE, JR.
Solicitor General

SANFORD M. LITVACK
Assistant Attorney General

LAWRENCE G. WALLACE
Deputy Solicitor General

HARRIET S. SHAPIRO
Assistant to the Solicitor General

ROBERT B. NICHOLSON
FREDERIC FREILICHER
Attorneys

Department of Justice
Washington, D.C. 20530
(202) 633-2217

JOSEPH F. NAKAMURA
Solicitor

THOMAS E. LYNCH
Associate Solicitor

United States Patent and Trademark Office
Washington, D.C. 20231

TABLE OF AUTHORITIES

	Page
Cases:	
Bergstrom, In re, 427 F.2d 1394	6
Howard, Ex parte, 328 Off. Gaz. Pat. Office 251, 1924 Dec. Com. Pat. 75	4
Parker v. *Flook*, 437 U.S. 584	5, 6, 7
Stubbs, In re, 58 F.2d 447	4
United States v. *Adams*, 383 U.S. 39	4
United States v. *New York Telephone Co.*, 434 U.S. 159	3
Waldbaum, In re, 457 F.2d 997	6
Statute and rule:	
35 U.S.C. 101	5, 6
Sup. Ct. R. 21.1a	3
Miscellaneous:	
H.R. Rep. No. 1923, 82d Cong., 2d Sess. (1952)	6
S. Rep. No. 1979, 82d Cong., 2d Sess. (1952)	6

In the Supreme Court of the United States

OCTOBER TERM, 1979

No. 79-855

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS, PETITIONER

v.

JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

REPLY BRIEF FOR THE PETITIONER

---

1. Respondents contend that they have created a "unique" machine (Resp. Br. 16), different in the arrangement or combination of its hardware components from any previous machine (Resp. Br. 10–12). They argue that the Patent and Trademark Office "ignor-[ed]" this combination taken as a whole, preferring to examine and characterize "each individual element except the hardware control unit (firmware) as old" (Resp. Br. 14). The Patent Office Board of Appeals,

however, specifically found that "save for the microprogramming" the hardware arrangement, as well as each component, including the control unit (see J.A. 169), was old (J.A. 200; see J.A. 152, 182). Respondents appealed the Board's finding that the hardware combination lacked novelty (J.A. 204 #2), but the Court of Customs and Patent Appeals did not disturb that finding (J.A. 18). Accordingly, respondents bear the burden here of showing that on the Patent Office record the Board's finding was clearly erroneous. This they have not done.

Respondents do not dispute that Coulter's and Carre's Figures 1-14i, describing the arrangement of hardware in their computer machines, are identical to the respondents' (see Pet. 8-9 n.12 and Pet. Br. 11 n.17; cf. Br. in Opp. 3, 19 and Resp. Br. 39-42). All that respondents have added is Figures 15a-c. They told the Patent Office that Figure 15a illustrated the switch system base instruction whose "operation code * * * is similar to the operation code of any instruction" (J.A. 78; see J.A. 29), and that Figures 15b and 15c were "flow charts" that disclosed a "microprogram" (J.A. 138, 155, 169).

Before this Court, however, respondents claim that Figures 15b and 15c disclose "additional hardware structure" (Resp. Br. 41). They refer specifically only to one such item, "hardware gate HG, reference numerals 1507 and 1514" (*ibid.*). According to respondents' specification this hardware gate is located in "auxiliary memory 1317a of the ALU 1317" (J.A. 79) and disclosed in Figure 13a (J.A. 112; see J.A. 63). But this gate was in fact part of the computer machine disclosed by Coulter and Carre. Thus, respondents' "aux-

iliary memory 1317a of the ALU 1317" (J.A. 79) is the same auxiliary memory identified in the Carre application (page 51, lines 24–27) and Coulter patent specification (col. 25, lines 39–41) as including "64 flip-flops [hardware gates] for storing miscellaneous states of the computer system." Identical language is used to describe auxiliary memory 1317a in respondents' specification (J.A. 63). Thus, the hardware gate identified by respondents in Figures 15b and 15c is merely one of the 64 flip-flops previously described by Carre and Coulter, and by respondents in Figure 13a of their own specification, and not "additional hardware structure" (Resp. Br. 41) revealed for the first time in Figures 15b and 15c.[1] The prior existence of an identical hardware arrangement also explains why, when respondents applied to the Patent Office, they did not say that their invention was a novel hardware arrangement but instead summarized it as: "A model-independent instruction in firmware [that] is provided

[1] Respondents contend (Resp. Br. 39–40) that the government's reliance on the Carre application and Coulter patent is improper under Rule 21.1a of the Rules of this Court, because that reliance raises issues concerning the novelty of their claimed invention not fairly included within the question presented in the petition. But it is respondents who have raised the issue of the novelty of their invention as an alternative ground for upholding the decision below (Resp. Br. 1, 13, 30, 33). Although it is permissible for them to do this without filing a cross-petition, see *United States* v. *New York Telephone Co.*, 434 U.S. 159, 166 n.8 (1977), they cannot legitimately object to the Commissioner's reliance, in response to that argument, on the patent applications of Coulter and Carre, of which they were concededly aware and which they acknowledged in their application (Resp. Br. 38–39; J.A. 25, 128). Indeed, they admitted to the patent examiner that the Coulter patent's "Figures 1–14h [*sic*, should be 14i] are identical" to respondents' corresponding figures (J.A. 136–137).

to change, upon execution, information in the system base which is located in both main memory and internal scratch pad registers" (J.A. 24; see J.A. 27).

2. Two of respondents' amici (Los Angeles Patent Law Ass'n Br. 9–10; Applied Data Br. 12–15), and perhaps respondents (Resp. Br. 33–34), further argue that the execution of respondents' novel microprogram in a conventional computer creates structural changes that result in a new machine. The running of the program supposedly does this by changing the computer's internal physical characteristics by, for example, enabling electronic switches to connect or disconnect various stuctures of the computer. Although patent law has recognized that a new machine might be created from old hardware elements (cf. *United States* v. *Adams*, 383 U.S. 39 (1966)), the Court has never recognized what in effect is a doctrine of transitory novelty. The mere execution of a novel microprogram in an old hardware combination creates no new physical structure. The hardware both before and after execution of the microprogram is the same. Only the electrical state of the circuitry changes during the running of the program. This change during the program run is thus transitory, not permanent, and does not turn an otherwise old machine into something that is new. See *In re Stubbs*, 58 F.2d 447 (C.C.P.A. 1932) (concrete pavement in uncured state not patentable); *Ex parte Howard*, 328 Off. Gaz. Pat. Office 251, 1924 Dec. Com. Pat. 75, 76 (1922) (claim to glass in molten state not patentable). Were it otherwise, a hand calculator would be patentable as a new machine each time a user placed a new constant into the calculator's memory, a telephone switching system would be pat-

entable as a new machine each time it diverted a call to a distant place over an alternative route not used before, and a television set would be patentable as a new machine each time it was tuned to a new channel not broadcasting when the television set was manufactured. Plainly, 35 U.S.C. 101 cannot be so read.[2]

3. Respondents and their amici contend that the Board has not correctly analyzed the claims.[3] They say that the Board erred by importing considerations of novelty into a Section 101 rejection (Resp. Br. 19, 22; American Patent Law Ass'n Br. 4; Halliburton Br. 8) and by identifying respondents' invention as a computer program when none was claimed (Resp. Br. 7; American Patent Law Ass'n Br. 6; Los Angeles Patent Law Ass'n Br. 8). The Board, however, followed *Parker* v. *Flook*, 437 U.S. 584, 594 (1978), in determining that the claims recited an unpatentable, albeit novel, algorithm for use in a digital computer and that, considered individually or as a whole, everything else

---

[2] It has been suggested that had respondents hard-wired their microprogram they could have patented their invention (Resp. Br. 37 n.106; Applied Data Br. 8-9; Los Angeles Patent Law Ass'n Br. 12-13; Halliburton Br. 5). Whether or not this is so, respondents are not claiming a hard-wired microprogram.

[3] To support their contention that the Patent Office has "misunderstood" what they have invented respondents refer to "substantive errors" made in the government's brief and petition (Resp. Br. 17; Resp. Br. App. 4a). These are supposedly "corrected," mostly without citation to authoritative sources, in appendices to respondents' main brief and brief in opposition to the petition (Resp. Br. App. 1a-4a; Br. in Opp. App. 1b-3b). But respondents' "corrections" merely evidence their disagreement with the government on the law, or on the conclusions to be drawn from the facts, and not on the facts themselves (*e.g.*, Resp. Br. App. 1a #1; Br. in Opp. App. 2b ¶¶ 2 & 3, 3b). We respond to the remaining technical points in the addendum to this brief.

in the claims was conventional (J.A. 202). Thus, respondents and their amici now seek to relitigate *Flook*.[4] They offer nothing, however, that was not previously argued in *Flook* and there is no reason to entertain their contentions for a second time in two years.[5] As for disclosure of a computer program, the examiner found that respondents' specification contained a "program implemented algorithm" (J.A. 151-152) and the Board concurred (*id.* at 202). These findings, which are confirmed by respondents' admissions (J.A. 141-142, 138, 155, 169), are amply supported in the record (see Pet. Br. 12-13 n.19; J.A. 27-28).

4. Finally, respondents (Resp. Br. 20-21) and their amici (Applied Data Br. 20-21; Los Angeles Patent

---

[4] The American Patent Law Association contends (Br. 24) that the Board has misread *Flook* by not recognizing that it was the "*generality* of the subsequent *application*" of the newly-discovered principle that made Flook's discovery unpatentable. But the Court understood that Flook's invention was designed for use in catalytic conversion processes (437 U.S. at 585-586) and that the claims did not "cover every conceivable application of the formula" (*id.* at 586). The instant invention is if anything even more general in applicability than was Flook's. Respondents themselves stressed in their application that their novel "instruction in firmware" was "model-independent" (J.A. 27) and thus was not limited to the particular digital computer machine described in their claims (see Pet. Br. 5, 19-20 n.29).

[5] For example, respondents' contention (Resp. Br. 24-26) that the provision in Section 101 that the claimed discovery must be "new" excludes only laws of nature, physical phenomena, and abstract ideas "because they have always existed," is inconsistent with the analysis in *Flook*, *supra*, 437 U.S. at 592-593. It is also inconsistent with the legislative history. S. Rep. No. 1979, 82d Cong., 2d Sess. 6 (1952); H.R. Rep. No. 1923, 82d Cong., 2d Sess. 7 (1952). See also *In re Bergstrom*, 427 F.2d 1394, 1401 (C.C.P.A. 1970); *In re Waldbaum*, 457 F.2d 997, 1003 (C.C.P.A. 1972).

Law Ass'n Br. 6; Halliburton Br. 3–4; American Patent Law Ass'n Br. 5) contend that for its own convenience the government here seeks a per se rule that all programmed machines, or even "computer related inventions" (Resp. Br. 21), are unpatentable. We have never contended, however, that a new digital computer, or a computer related invention, is unpatentable when it involves an inventive novelty in its structure or in its type or arrangement of hardware. We do ask this Court to reaffirm that if the only novelty in a claimed digital computer or other claimed invention consists of a programmed algorithm there is no patentable invention. *Parker* v. *Flook*, *supra*, 437 U.S. at 594.

For the reasons stated herein and in our main brief, the judgment of the Court of Customs and Patent Appeals should be reversed.

Respectfully submitted.

WADE H. MCCREE, JR.
Solicitor General

SANFORD M. LITVACK
Assistant Attorney General

LAWRENCE G. WALLCE
Deputy Solicitor General

HARRIET S. SHAPIRO
Assistant to the Solicitor General

ROBERT B. NICHOLSON
FREDERIC FREILICHER
Attorneys

JOSEPH F. NAKAMURA
Solicitor

THOMAS E. LYNCH
Associate Solicitor
United States Patent
and Trademark Office

OCTOBER 1980

ADDENDUM

1. Contrary to petitioner's statement (Pet. 5 n.7; Pet. Br. 4 n.7, 20 n.30), respondents maintain that the PROM described in their specification ("a 9K bipolar integrated circuit programmable read-only memory (PROM) mixed with read/write random access store (RAM)" (J.A. 61)) is not reprogrammable (Resp. Br. App. 1a–2a #2; see Br. in Opp. 1b ##1, 2). But memory units of the same bipolar type as respondents' that are capable of being reprogrammed in the field were patented and in use before respondents' application was filed. See, *e.g.*, U.S. Patent 3,662,351 to Ho, *et al.*, patented May 9, 1972, filed Mar. 30, 1970 (abstract and col. 6, line 60); U.S. Patent 3,717, 852 to Abbas, *et al.*, patented Feb. 20, 1973, filed Sept. 17, 1971 (abstract and col. 3, lines 1–9; col. 8, lines 38–40). And see Ho and Maley, "Latent Image Can Provide Chips With Built-In Control Memories," *Electronics* 82–83 (Aug. 16, 1971). Respondents do not say why they have included in their specification hardware of a type that permits such reprogramming if their PROM cannot be reprogrammed (see Pet. Br. 20 n.30). Moreover, although we twice noted that respondents' specification claims their PROM can be "reloaded" (J.A. 60; Pet. 5 n.7; Pet. Br. 20 n.30), respondents have failed to explain to the Court what this means if it does not mean that their PROM can be reprogrammed in the field.

2. The switch system base instruction is not, as respondents contend, "hardware incorporated into respondents' computer" (Resp. Br. App. 2a–3a #3; cf. Pet. Br. 4–5 & n.8). Respondents told the Patent Of-

fice that the instruction includes an operation code which "is similar to the operation code of any instruction and indicates the operation to be performed" (J.A. 78).

3. Respondents (Resp. Br. App. 2a–3a, ## 3, 4) have interpreted the word "user" in petitioner's brief (at 2–5) to mean only the software programmer who seeks to get data out of the computer (*i.e.*, the applications programmer). We meant "user" to include the systems programmer who writes programs regulating the computer's internal operations.

☆ U.S. GOVERNMENT PRINTING OFFICE: 1980 329439 92

Supreme Court, U. S.
FILED
OCT 8 1980
MICHAEL RODAK, JR., CLERK

NO. 79-855

IN THE

Supreme Court of the United States

OCTOBER TERM, 1980

SIDNEY A. DIAMOND,
COMMISSIONER OF PATENTS AND TRADEMARKS,
Petitioner,

v.

JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN,
Respondents.

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

REPLY BRIEF FOR THE RESPONDENTS

NICHOLAS PRASINOS
Honeywell Information Systems Inc.
Office of the General Counsel
200 Smith Street
Waltham, Massachusetts 02154
(617) 895-6129
Counsel for Respondents

Of Counsel:
FAITH F. DRISCOLL
HENRY L. HANSON
LAWRENCE J. MARHOEFER
RONALD T. REILING

Washington, D.C. • CLB PUBLISHERS' • LAW PRINTING CO. • (202) 393-0625

IN THE

Supreme Court of the United States

OCTOBER TERM, 1980

NO. 79-855

SIDNEY A. DIAMOND,
COMMISSIONER OF PATENTS AND TRADEMARKS,
Petitioner,

v.

JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN,
Respondents.

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

REPLY BRIEF FOR THE RESPONDENTS

TABLE OF AUTHORITIES

Page

Cases:

Diamond v. Chakrabarty,
____ U.S. ____ , Opinion No. 79-136 (1980)..... 4

Gottschalk v. Benson,
409 U.S. 63, 93 S. Ct. 253 (1972).....4,6

Parker v. Flook,
437 U.S. 584, 98 S. Ct. 2522 (1978) 4,6

Statutes:

35 U.S.C. 101.....*Passim*
35 U.S.C. 102.....*Passim*
35 U.S.C. 103.....*Passim*

The Respondents' Reply Brief is necessitated by the mischaracterization of the issue before this Court by the Commissioner in his Reply Brief for the Petitioner.

The Commissioner in his Reply Brief has changed his position from that taken in his Brief for the Petitioner, has misrepresented the Respondents' position, and has misconstrued the Respondents' invention. Notwithstanding this obfuscation by the Commissioner, the issue before the Court is whether the Respondents' invention is a machine as that term is used in 35 U.S.C. 101. The issue is *not* the novelty of the Respondents' invention under 35 U.S.C. 102 or 35 U.S.C. 103.

A. THE COMMISSIONER HAS CHANGED HIS CHARACTERIZATION OF RESPONDENTS' INVENTION IN HIS REPLY BRIEF FROM THAT PRESENTED IN HIS BRIEF FOR THE PETITIONER.

On page 5 of his Brief for the Petitioner the Commissioner characterizes Figures 15b and 15c as flow charts for a microprogram, yet on pages 2 and 3 in his Reply Brief, the Commissioner changes his position and argues that Figures 15b and 15c include hardware previously described by the Carre application and the Coulter patent. The disclosures of Carre or Coulter *do not* include Figures 15a, 15b, and 15c and the accompanying description included in Respondents' application. The Commissioner, therefore, is incorrect in alleging that the Respondents' hardware invention is disclosed in Carre and Coulter.

B. THE COMMISSIONER STILL HAS NOT DEFINED OR IDENTIFIED ANY PROGRAM IN THE RESPONDENTS' APPLICATION BUT INSISTS ON CALLING THE "RULE OF ACTION OF THE MACHINE" A PROGRAM.

All machines have a rule of action. (See Brief for The Respondents, pg. 12, footnote 40). The rule of action is embodied in hardware. A simple example will make this clear.

Take an ordinary electric circuit comprising only a light bulb and electric switch. Certainly, this is the type of device that is eligible for patent protection under 35 U.S.C. 101.

Let us now add a cam which engages the light switch, and a motor for turning that cam at predetermined speeds. The cam engages the light switch to turn the light on and off as it rotates.

We now have an entirely new machine - a stroboscope. The rotation of the cam can be set to turn the light on and off at the speed of a bullet fired from a gun. When this is done, the stroboscope "freezes" the motion of the bullet and a clear picture of it can be taken. (Professor Edgerton of MIT has made many such pictures using a stroboscope.)

The rule of action of this machine is embodied in the cam. The cam may be likened to firmware in a computer. As a matter of fact, the cam in this stroboscope could be replaced by firmware. Can it now be said that the stroboscope is not eligible for patent protection merely because firmware has replaced the cam? The answer is obvious - it is the type of invention that the patent laws were enacted to protect — it is a machine as that term is used in 35 U.S.C. 101.

The Respondents' invention although infinitely more complex is similar in that the rule of action is embodied in the claimed hardware combination.

C. THE COMMISSIONER MUST CITE PRIOR ART UNDER 35 U.S.C. 102 OR 35 U.S.C. 103 TO SHOW IN WHAT RESPECTS A HARDWARE ARRANGEMENT IS OLD.

There is no issue of novelty under 35 U.S.C. 102, 103 present in this case as alleged by the Commissioner. Even if such issue were present the Commissioner is merely relying on a bare allegation on page 2 of his Reply Brief to support his position that the Respondents' hardware combination is old. He has cited no prior art on the record to show that "the hardware arrangement, as well as each component" is old nor has he shown how such components cooperate and function with each other to produce the improved result of Respondents' invention. Allegations are not "findings" as stated by the Commissioner. The Commissioner is clearly erroneous in stating that an allegation is a finding.

Despite the fact that there is no issue of novelty under 35 U.S.C. 102, 103 before this Court the Commissioner is using the Respondents' own disclosure to allege that Respondents' invention is old under 35 U.S.C. 102 or 35 U.S.C. 103. The Commissioner attempts this by reconstructing the Respondents' invention to fit within the disclosures of Carre and Coulter. The Commissioner by hindsight is utilizing the Respondents' invention to infer the disclosure of a gate in Coulter. Even if such a gate were present in Coulter and Carre (which is not admitted by the Respondents), there is no disclosure in Carre or Coulter to show how it cooperates and functions with other hardware elements in a similar manner as the Respondents' invention to achieve the improved results. Such hindsight reconstruction of the invention and the use of the Respondents' own disclosure as prior art is clearly erroneous and contrary to established law.

D. THE COMMISSIONER IMPLIEDLY ADMITS THAT HE IS UTILIZING NOVELTY REQUIREMENTS TO REJECT COMPUTER RELATED INVENTIONS UNDER 35 U.S.C. 101.

In answer to an amicus statement that the Government for its own convenience seeks a per se rule that computer related inventions are unpatentable the Commissioner replied as follows:

> "We have never intended, however, that a new digital computer, or a computer related invention, is unpatentable when it involves an inventive novelty in its type or arrangement of hardware."

Novelty is a 35 U.S.C. 102 or 103 requirement and not a 35 U.S.C. 101 requirement. Consistent with this Court's decisions, Flook, Benson and Chakrabarty, 35 U.S.C. 101 requires a man-made machine or improvement thereof which has not always existed. When these criteria are met the invention is then statutory subject matter under 35 U.S.C. 101 and is eligible for examination of novelty under 35 U.S.C. 102 and 103. Respondents' invention meets these criteria.

E. THE COMMISSIONER IN ATTEMPTING TO IMPORT 35 U.S.C. 102 AND 103 REQUIREMENTS TO SHOW THE SIMILARITY BETWEEN THE COULTER PATENT AND THE BRADLEY INVENTION ADMITS THAT THE INVENTION IS PATENTABLE SUBJECT MATTER UNDER 35 U.S.C. 101.

It is improper to import 35 U.S.C. 102 and 103 prior art factual considerations into 35 U.S.C. 101. Both the Carre

application, U.S. Serial Number 528,951 and the Coulter patent, U.S. Patent Number 3,891,874, issued June 24, 1975, were called to the attention of the Commissioner during prosecution by the Respondents. (See J.A. p. 25, Ref. 15 and J.A. p. 136.) The Commissioner chose not to cite these documents under 35 U.S.C. 102 or 103, and thereby precluded the establishment of a factual record regarding issues of novelty and prior invention.

The Respondents stand by their argument in the Brief for Respondents, pps. 38-42. The Commissioner by alleging that the Respondents' invention is similar to the invention of Coulter and Carre is admitting thereby that Respondents' invention is the type that the Patent Laws were enacted to protect. The invention is therefore a machine as that term is used in 35 U.S.C. 101.

F. THE COMMISSIONER HAS MISREPRESENTED THE RESPONDENTS' POSITION.

The Commissioner has erroneously misrepresented the Respondents' position to this Court (refer to Reply Brief of the Petitioner, p. 4, paragraph 2, lines 3-6). Contrary to the Commissioner's misrepresentation, the Respondents are on record as indicating that a new program in a conventional computer does not create a new machine. (Refer to Respondents' Brief in Opposition to a Writ of Certiorari to the United States Court of Customs and Patent Appeals, p. 9, first full paragraph). This is not an issue.

The Commissioner additionally misrepresents the Respondents' position in response to Respondents' statement that the introduction of the Carre application and the Coulter patent is improper under Rule 21.1a of the Rules of

this Court. The Commissioner now alleges that it is the Respondents who have raised the issue of novelty of their invention (refer to Footnote 1, page 3 of the Reply Brief for the Petitioner). This is not true. Furthermore, there is no basis for this position in those portions of Respondents' Reply Brief cited by the Commissioner. The correct facts are that the court below stated that issues of novelty and non-obviousness were not before it (refer to J.A. 18). It agreed with respondents in that their invention is a machine or apparatus, and that the claims fall "literally within the boundaries of 35 U.S.C. 101." (J.A. 18).

In view of the established record, the Commissioner's basis for attempting to reintroduce the question of novelty is inaccurate, without support and improper under the Rules of this Court (i.e., Rule 21.1a).

G. THE COMMISSIONER HAS ADMITTED THAT HE WANTS THIS COURT, *NOT* CONGRESS TO DECLARE A PER SE RULE THAT COMPUTER PROGRAMS AND COMPUTER RELATED INVENTIONS ARE NOT PATENTABLE.

The Commissioner by his comments in his Reply Brief (Commissioner's Reply Brief, paragraph 4, pp. 6-7) has tacitly admitted that for his own convenience he seeks a per se rule that computer programs and computer related inventions are unpatentable. As this Court has unequivocally stated in both *Benson* and *Flook,* such policy issues are for Congress to decide - not the Court.

For the reasons stated herein and in Respondents' main Brief, the judgement of the Court of Customs and Patent Appeals should be affirmed.

Respectfully submitted,

NICHOLAS PRASINOS
Honeywell Information Systems Inc.
200 Smith Street
Waltham, Massachusetts 02154
(617) 895-6129

Counsel for Respondents

Of Counsel:
FAITH F. DRISCOLL
HENRY L. HANSON
LAWRENCE J. MARHOEFER
RONALD T. REILING

FILED
JUL 28 1980
MICHAEL RODAK, JR.,

No. 79-855

IN THE

Supreme Court of the United States

October Term, 1979

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS, *Petitioner*

v.

JOHN J. BRADLEY and BENJAMIN S. FRANKLIN

On Writ of Certiorari to the U. S. Court of Customs and Patent Appeals

BRIEF AMICUS CURIAE FOR THE AMERICAN PATENT LAW ASSOCIATION, INC. IN SUPPORT OF AFFIRMANCE AND/OR REMAND

AMERICAN PATENT LAW ASSOCIATION, INC.
2001 Jefferson Davis Highway
Arlington, Virginia 22202
DONALD R. DUNNER, President

KENNETH E. KUFFNER
ARNOLD, WHITE & DURKEE
2100 Transco Tower
Houston, Texas 77056
Attorney

Of Counsel:
TRAVIS GORDON WHITE
JOHN D. NORRIS
ARNOLD, WHITE & DURKEE
2100 Transco Tower
Houston, Texas 77056

Alpha Law Brief Co., One Main Plaza, No. 1 Main St., Houston, Texas 77002

TABLE OF CONTENTS

	Page
I. AUTHORITY TO FILE	1
II. THE APLA AS A FRIEND OF THE COURT IN THE PUBLIC INTEREST	1
A. The Nature of the Amicus Organization	1
B. The Interest of Amicus is the Public Interest in Preservation of the Patent System	2
C. APLA Takes No Position on the Patentable Merits of the Invention Involved in This Application	2
III. PURPOSE OF THIS BRIEF	4
IV. COMMENT ON THE GOVERNMENT'S STATEMENT OF THE ISSUES BEFORE THE COURT	6
V. ARGUMENT	7
A. It Is Only Through the Use of an Ill-Conceived, Irrational Approach to Patent Claim Analysis That This Case Can be Said to Present the Issue of Whether Computer Programs Are Statutory Subject Matter Eligible for Patent Protection	7
B. The Government's Two-Step Approach to Determination of Whether Claimed Subject Matter is Eligible for Patent Protection Violates Statutory Mandates and Prior Case Law Requirements on Claim Interpretation	13
C. Dictum in *Parker v. Flook* Cited in Support of the Two-Step Approach Cannot be Justified	15
D. The Decisions in *Benson* and *Flook* Can Reasonably Be Urged to Be Consistent with a Rational Philosophy Found in Past Decisions of this Court Which Supports the Constitutional Goals of Our Patent System But *Does Not* Support the Artificial Analysis Proposed by the Government	21
E. Conclusion	26

TABLE OF AUTHORITIES

CONSTITUTION

	Page
U. S. Const., art. 1, § 8, cl. 8	5

FEDERAL STATUTES AND REGULATIONS

28 U.S.C. § 1542 (1976)	3
35 U.S.C. § 101 (1976)	*passim*
35 U.S.C. § 102 (1976)	4, 8, 12, 28
35 U.S.C. § 103 (1976)	4, 8, 12, 19, 28
35 U.S.C. § 112 (1976)	4, 6, 14, 28
35 U.S.C. § 131 (1976)	14
35 U.S.C. § 144 (1976)	3
37 C.F.R. § 1.104 (1979)	8
37 C.F.R. § 1.105 (1979)	8, 9

CASES

In re Alul, 468 F.2d 939 (CCPA 1972)	3
In re Arkley, 455 F.2d 586 (CCPA 1972)	3
In re Bergy, 596 F.2d 952 (CCPA 1979)	3
In re Bernhart, 417 F.2d 1395 (CCPA 1969)	11
In re Bradley, 600 F.2d 807 (CCPA 1979)	6
Brenner v. Manson, 383 U.S. 519 (1966)	25
In re Crowell, 84 F.2d 206 (CCPA 1936)	14
Dann v. Johnston, 425 U.S. 219 (1976)	8
In re Deutsch, 553 F.2d 689 (CCPA 1977)	14
Diamond Rubber Co. v. Consolidated Rubber Tire Co., 220 U.S. 428 (1911)	19
In re Fischer, 113 F.2d 492 (CCPA 1940)	14
In re Gelnovatch, 595 F.2d 32 (CCPA 1979)	13
Gottschalk v. Benson, 409 U.S. 63 (1972)	*passim*
Graham v. John Deere Co., 383 U.S. 1 (1966)	12
In re Kebrich, 201 F.2d 951 (CCPA 1953)	14
In re Krichbaum, 39 F.2d 280 (CCPA 1930)	14
McClain v. Ortmayer, 141 U.S. 419 (1891)	13
In re Moore, 429 F.2d 985 (CCPA 1970)	14
O'Reilly v. Morse, 56 U.S. (15 How.) 62 (1853)	22
Parker v. Flook, 437 U.S. 584 (1978)	*passim*
In re Phillips, 608 F.2d 879 (CCPA 1979)	8
In re Prater, 415 F.2d 1393 (CCPA 1969)	15
In re Priest, 582 F.2d 33 (CCPA 1978)	14
In re Sarkar, 588 F.2d 1330 (CCPA 1979)	8
U. S. v. Adams, 383 U.S. 39 (1966)	16, 17

III

SECONDARY AUTHORITIES

	Page
Manual of Patent Examining Procedure, § 707.07(g) (4th ed. 1979)	8, 10
SENATE COMM. ON COMMERCE, SCIENCE, AND TRANSPORTATION, 96TH CONG., 1ST SESS., REPORT ON NATIONAL STRATEGY FOR TECHNOLOGICAL INNOVATION (Comm. Print 1979)	20

No. 79-855

IN THE

Supreme Court of the United States

October Term, 1979

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS, *Petitioner*

v.

JOHN J. BRADLEY and BENJAMIN S. FRANKLIN

On Writ of Certiorari to the U. S. Court of Customs and Patent Appeals

BRIEF AMICUS CURIAE FOR THE AMERICAN PATENT LAW ASSOCIATION, INC. IN SUPPORT OF AFFIRMANCE AND/OR REMAND

I. AUTHORITY TO FILE

This amicus brief is presented under Rule 36. The letters of consent are on file with the Clerk.

II. THE APLA AS A FRIEND OF THE COURT IN THE PUBLIC INTEREST

A. The Nature of the Amicus Organization

The American Patent Law Association, Inc. ("APLA") is a non-profit membership organization of over 4,000

lawyers interested in intellectual property law, that is, patents, trademarks, copyrights, trade secrets and related areas of law. Members of APLA not only practice before the Patent and Trademark Office ("PTO"), but also are concerned with patent litigation, with counseling on infringement and validity matters and with licensing or other transfer of patented and non-patented technology.

APLA membership is not restricted to any segment of the profession. Attorneys in private practice and those employed by corporations, universities and government are included. Members of the association and their clients represent widely divergent interests and views. They represent individuals, organized research and development (both academic and commercial), and large and small industrial and business enterprises.

B. The Interest of Amicus is the Public Interest in Preservation of the Patent System

The APLA has expertise in the day-to-day workings of, and the public benefits derived from, the American patent system. Since the broadly based association membership is not divided into plaintiffs' and defendants' bars, APLA is uniquely qualified to speak fairly on the general issues which affect the public interest in preserving the patent system.

C. APLA Takes No Position on the Patentable Merits of the Invention Involved in This Application

The jurisdiction of the Court of Customs and Patent Appeals ("CCPA"), in its review of the decisions of the

PTO tribunals, is limited.[1] This means that the decision of the CCPA in the case presently under review must *NOT* be considered as a conclusion that a patent should issue.[2] The CCPA decided *ONLY* that the PTO was wrong in rejecting all claims in the application for being outside the categories of subject matter eligible for patent protection[3] as set forth in 35 U.S.C. § 101 (1976). No

---

1. While 28 U.S.C. § 1542 states broadly that the CCPA "shall have jurisdiction of appeals from decisions of" the Patent and Trademark Office, the patent statute requires that the court "shall hear and determine such appeal on the evidence produced before the Patent and Trademark Office, *and the decision shall be confined to the points set forth in the reasons of appeal.*" 35 U.S.C. § 144 (1976) (emphasis added). The policy of the CCPA in this regard is well summarized in the following passage from *In re Alul*, 468 F.2d 939, 941 (CCPA 1972):

> We note that a substantial part of the solicitor's brief is devoted to arguments which were not raised below and which are tantamount to new grounds of rejection. . . . We will not consider these new arguments. We realize the position this leaves appellants in, but we are confident that if any merit were found in the new arguments in cases like this, the Patent Office would reopen prosecution to assert them and give appellants a chance to respond to them.

2. "It is the Patent Office which grants patents, not this Court." *In re Arkley*, 455 F.2d 586, 589 (CCPA 1972).

3. It is critical that the Court understand the distinction we make in this brief between the issue of whether claimed subject matter *is eligible for patent protection*, on the one hand, and the issue of whether claimed subject matter *satisfies the conditions for patentability, i.e.*, is new, useful and nonobvious, and thereby merits the grant of a patent, on the other hand. It is the former issue only which is before the Court because that is the only ground on which the PTO chose to finally reject the application here involved. The APLA agrees that inquiry as to either issue must consider the claimed subject matter *in its entirety*, but emphasizes that each inquiry must be made separately and independently of the other. *Cf. In re Bergy*, 596 F.2d 952, 959 (CCPA 1979), wherein the CCPA expressed its concern regarding the

> commingling of distinct statutory provisions which are conceptually unrelated, namely, those pertaining to the *categories* of inventions in § 101 which *may* be patentable and to the *conditions* for patentability demanded by the statute for inventions

other issues were before the CCPA. The PTO chose not to pursue any other ground for rejection. The court thus could not consider questions related to the novelty or nonobviousness of the claimed subject matter.

The APLA believes that value judgments as to whether these respondents should or should not ultimately be *entitled to* a patent[4] have no proper place in deciding whether the subject matter which respondents seek to have patented is *within the statutory categories which are eligible for* the patent grant. While taking no position on whether the application now before the Court satisfies the various statutory requirements and conditions for patentability,[5] the APLA fully supports the holding of the CCPA that the claims of this application define subject matter which is eligible for patent protection under § 101 of the statute.

III. PURPOSE OF THIS BRIEF

The public interest in an effective patent system demands a broader view of the issues than that provided by the petitioner, which appears to be reflective only of administrative problems in the PTO.

---

within the statutory categories, particularly the nonobviousness condition of § 103.

4. We are concerned that the nature of the Bradley and Franklin invention, coupled with the manner in which the government has developed its argument, may persuade some justices to vote for reversal *only* because they have become convinced that a patent ought not to issue on this particular application. Alternatively, some might conclude that to affirm the ruling of the CCPA will result in the issuance of many patents where the only novelty of the invention involved lies in the use of a programmed computer. We urge the Court to be clear that an affirmance of a CCPA decision does not mandate that even the application which is before the Court issue as a patent.

5. 35 U.S.C. §§ 102, 103 and 112 (1976), in particular.

The government seeks in this case and in its companion *Diamond v. Diehr* (No. 79-1112), as it did in *Gottschalk v. Benson,* 409 U.S. 63 (1972), and *Parker v. Flook,* 437 U.S. 584 (1978), to have the Court foreclose the granting of patents for all inventions involving computer programming. This attempt is at odds with the constitutional purpose behind our patent system, *i.e.,* "to promote the Progress of . . . the useful Arts."[6] In this technology, as in all others, it is a socially desirable goal, effectively implemented by our patent system, to induce and to encourage by patent grant the disclosure of new and useful information and the commitment of capital to research and development.

Moreover, the government's brief urges an approach to claim analysis—having potential application beyond the specific area of computer program-related inventions —which could severely undermine the patent system itself.

It is a major purpose of this brief to persuade the Court that this approach cannot be supported in law or in logic. The CCPA in this case was correct in not following the approach urged by the PTO. The decision of the CCPA (contrary to the assertions of the government and certain amici) is not inconsistent with the holdings in *Benson* and *Flook*. Rather, it is based upon a reasoned approach,[7] consistent with those cases and with the aims of the patent system, whereby claimed

---

6. U.S. Const. art. 1, § 8, cl. 8.

7. The CCPA approach and how it is fully consistent with the decisions in *Benson* and *Flook* and with the philosophy underlyinng those decisions is fully discussed in our brief in *Diamond v. Diehr,* No. 79-1112, filed concurrently herewith.

subject matter involving computer-related or mathematics-related concepts may be tested to determine whether it is eligible for patent protection.

IV. COMMENT ON THE GOVERNMENT'S STATEMENT OF THE ISSUES BEFORE THE COURT

The Petition presents the question as:

> Whether a computer program that regulates the internal operation of a computer is patentable subject matter under 35 U.S.C. 101.

This suggests that respondents, in the application before the Court, are seeking to patent a computer program, but this could not be further from the truth. We ask the Court to compare "the question" presented with the actual language of the application claims. Respondents here are not claiming a "computer program." In fact, respondents have pointed out not only that they are *not* claiming a computer program, but also that they have never intended to do so.[8] The CCPA, in reversing the decision of the PTO, purposefully avoided the approach to claim analysis urged by the PTO and expressly determined that the claimed invention was "a combination of tangible hardware elements—a machine—including some hardware elements which contain microprogrammed information termed 'firmware'." *In re Bradley*, 600 F.2d 807, 813 (CCPA 1979).

---

8. For more than 100 years the patent statutes have required each patent applicant to "particularly point out and distinctly claim the subject matter he regards as his invention." This requirement continues in the present statute in the second paragraph of 35 U.S.C. § 112 (1976).

It is a fact, however, that respondents' invention *does require,* in its practical application, the *use* of a programmed computer and the claims now before the Court make reference to this fact. The question presented by the government focuses on this point alone, through an exercise which ignores what the patent applicants are seeking to patent and what the court below found to be defined by the application claims. The "how" and "why" of this exercise, its basic impropriety, as well as its insidious danger to the constitutional underpinnings of our patent system, will be the principal focus of this brief.

V. ARGUMENT

A. It Is Only Through the Use of an Ill-Conceived, Irrational Approach to Patent Claim Analysis That This Case Can be Said to Present the Issue of Whether Computer Programs Are Statutory Subject Matter Eligible for Patent Protection

As the government states it, the "Question Presented" in this case and in its companion, *Diamond v. Diehr,* is whether a particular computer program is patentable subject matter under § 101. This Court has already declined twice to rule broadly on this issue, indicating that it should be resolved by Congress.[9] Moreover, as we earlier mentioned, a computer program, per se, is not claimed as the invention.

Nevertheless, the PTO, and now the Solicitor General, argue that the subject matter of these claims must be

---

9. *Gottschalk v. Benson, supra,* 409 U.S. at 72-73; *Parker v. Flook, supra,* 437 U.S. at 595.

considered as nothing more than a computer program. Why do they do so? The answer, we believe, is that the government seeks judicial approval of a proposed (perhaps already established)[10] PTO practice, against its stated policy,[11] whereby all claims in computer-related cases would be summarily rejected under § 101 as ineligible for patent protection in a preliminary, "threshold" examination procedure which does not reach the merits of the claimed invention.

Quite literally, the PTO, in these appeals, is attempting to stem a tide, for the computer has become the life-blood tool for technological development. Because of its inherent capabilities—its speed, its computational power, its ease of reconfiguration as system requirements change —it has found applicability in every field of technology. The advent of the microprocessor (a scaled down, "mini" computer) is having the same effect that the transistor had when it became a basic tool for designers. As a result, the PTO is being asked to examine an ever-increasing number of applications, in many varied areas of complex

---

10. In both cases now before the Court, rejections based upon other statutory grounds were initially made but abandoned without comment before the appellate procedures began. In the years following this Court's decision in *Dann v. Johnston,* 425 U.S. 219 (1976), there has been such a notable absence, in the "computer program" cases brought before the CCPA of issues other than those relating to threshold questions that that court has felt compelled to express its concern. Witness the court's footnote comment in *In re Phillips,* 608 F.2d 879, 883n.6 (CCPA 1979): "The PTO has apparently not considered appellants' application for compliance with 35 U.S.C. §§ 102 and 103. Piecemeal examination is to be avoided. 37 CFR 1.104 and 1.105. *See also* MPEP 707.07(g)." The CCPA has also stated that it will not "assume de novo the function of an examiner who has declined, in reliance upon § 101, to conduct an examination for patentability under §§ 102 and 103." *In re Sarkar,* 588 F.2d 1330, 1334n.10 (CCPA 1979).

11. *See* note 16 *infra* and accompanying text.

technology, wherein subject matter sought to be patented involves the *use* of computer programming or programmed data processors.[12]

Moreover, whereas in *Benson* the invention claimed was held to be the mathematical algorithm itself, and in *Flook* the claimed invention was held to be a mathematical manipulation,[13] many new inventions for which patents are being sought are computer program-related *only* in that a program algorithm, or the use of a programmed data processor, is involved in a *portion* of the claimed subject matter. Since such claimed subject matter, as a whole, is neither a mathematical algorithm nor a general mathematical manipulation, the *Benson* and *Flook* decisions, by themselves, do not justify abandonment of established PTO policy requiring that every application be given as complete an examination as possible.[14]

---

12. We have no reason to doubt the unsubstantiated allegation made in the government's petition in this case that "[t]he Patent and Trademark Office presently has pending more than 3000 patent applications in which the patentability of computer software or firmware is a potential issue." Pet. at 13. To place this in proper perspective, that figure represents about 1% of the total number of applications for patent which are now pending.

13. In *Flook,* the mathematical algorithm was presented as an equation, which was then solved.

14. Of the more than 100,000 applications considered annually by the PTO, many could be summarily rejected on certain "threshold" matters without giving any serious attention to the questions of novelty and obviousness of the application subject matter. More often than not, however, the examiner's conclusion on a threshold question is a preliminary one, made in an area where there is much room for dispute. Accordingly, it is PTO policy not to suspend the continued examination of the patentable merits of the invention until the threshold question is resolved, but rather to render a complete examination so as to avoid "piecemeal" consideration of the application. PTO Rule 105, 37 C.F.R. § 1.105 (1979), expressly states that an examiner's action on an application should be "complete as to all matters, except . . .

Accordingly, the PTO is in need of either an increased appropriation from Congress by which it could handle this increasing workload, or the authority from this Court by which it may abdicate its most important modern-day function by refusing even to consider the patentable merits of computer program-related inventions. Quite clearly, the PTO has chosen to seek the latter. The government's statement of the question in each of these cases is merely the culmination of a PTO effort to force the facts of the case into the issue which it wishes decided, that is, that *any* computer program-related invention must be regarded as outside the statutory categories of subject matter considered eligible for patent protection.

Considered in its entirety, the government's approach to considerations under § 101 involves the following two-step test:

(1) Dissect the claim to identify the point of novelty, and

(2) If the point of novelty itself does not fit into a class of patentable subject matter, and the claim considered without the novel part is old, then the claim is non-statutory.

If the "point of novelty" involves the use of a computer program, the government reads *Benson* as controlling in two different respects. First, it says that *Benson* stands for the non-patentability of *all* computer programs. Second, it contends that the holding in *Benson* —that mathematical algorithms are not patentable sub-

---

in appropriate circumstances." The Manual of Patent Examining Procedure ("MPEP") states that "the examiner ordinarily should reject each claim on all valid grounds available." MPEP § 707.07(g) (4th ed. 1979).

ject matter—must also extend to programmed computers since programmed computers operate according to simple mathematical rules. Accordingly, it concludes that all programmed computers are non-patentable subject matter.

The second part of the test, by which the remainder of the claim, after it has been determined to be old, can be ignored and the "point of novelty" considered by itself to determine whether the claim as a whole defines statutory subject matter,[15] has been derived by the government[16] from the majority opinion in *Parker v. Flook.*

The ALPA believes that this two-step approach cannot be supported either under the patent statute or under clearly applicable case law. As we will show,

(1) The test cannot withstand an analysis of its premises;

(2) Its artificiality will result in unreasonable discrimination and disincentives which are at odds with the patent system; and

---

15. This is a revival of an approach considered and rejected by the CCPA more than ten years ago. At that time, the PTO proposed to reject as non-statutory any claim wherein the "point of novelty" could be characterized as a "mental step." The CCPA's comments in treating this question are particularly appropriate:

> A much closer question arises, however, when we consider the *principle* extracted by the examiner from the mental step cases and by the board from the printed matter cases. The principle may, we think, be fairly stated as follows: If, in an invention defined by a claim, the novelty is indicated by an expression which does not itself fit in a statutory class . . ., then the whole invention is non-statutory since all else in the claim is old. We do not believe this view is correct under the Patent Act and the case law thus far developed.

In re Bernhart, 417 F.2d 1395, 1399 (CCPA 1969).

16. *See* Pet. Br. in *Diamond v. Bradley* at 9; Pet. Br. in *Diamond v. Diehr* at 7-10.

(3) It is not even consistent with the rationale supporting the two Supreme Court cases from which it is supposed to be derived.

The issue of whether claimed subject matter fits into one of the § 101 statutory categories which Congress declared eligible for patent protection should properly be considered before the patentable merits of the invention are examined. In *Parker v. Flook,* this Court stated:

> The obligation to determine what type of discovery is sought to be patented must precede the determination of whether that discovery is, in fact, new or obvious.

437 U.S. at 593. It is not this premise with which we take issue.

The APLA does take strong issue, however, with any examination procedure which does not consider the claimed subject matter *as a whole.*[17] We cannot abide the approach utilized in the two cases now before the Court, and especially the two-step "test" urged by the government. We agree fully with the views of Chief Judge Markey:

---

17. We *do not* contend that it is necessarily wrong to analyze a patent claim element by element. The Court in *Graham v. John Deere Co.,* 383 U.S. 1, 17 (1966), made it clear that the statutory conditions for patentability must be tested by an analysis which determines the differences between the subject matter defined by the claim and that which is in the prior art. We *do* contend, however, that a § 101 inquiry into subject matter patentability must consider the claimed "subject matter as a whole." The Court, in *Flook,* accepted this position. 437 U.S. 584, 594n.16.

It would be absurd to determine whether a patent claim meets the conditions for patentability stated in 35 U.S.C. §§ 102 and 103 by any analysis which ignores portions of the claim language. We submit that it can be no less absurd to determine whether claimed subject matter is eligible for patent protection under § 101 when substantial portions of the claim are ignored.

> The advent of a wholly new technology confronts the Patent and Trademark Office (PTO) with administrative problems in performing its vital service to the public interest in encouraging true progress of the useful arts. The solution to administrative problems does not lie, however, in so interpreting the law as to reduce an administrative burden. Avoiding a petition to Congress for increased funds or legislative relief, using § 101 to exclude new technology, presenting to the Supreme Court patentability issues not considered in the PTO or in this court, and employing phrases from the Court's opinions as rejection rubrics under § 101 in subsequent cases, is an approach long overdue for abandonment.

In re Gelnovatch, 595 F.2d 32, 42 (CCPA 1979) (Markey, C.J., dissenting).

It is only through the use of the government's contrived and irrational "two-step" approach to claim analysis that the issue of whether computer programs are statutory subject matter eligible for patent protection can be argued to exist in this case. Without the two-step approach, the CCPA decisions in both these companion cases can be affirmed, and sent back to the PTO for proper examination, without the need for any discussion of the issue of whether "computer programs," in and of themselves, are proper subject matter for the patent grant.

B. The Government's Two-Step Approach to Determination of Whether Claimed Subject Matter is Eligible for Patent Protection Violates Statutory Mandates and Prior Case Law Requirements on Claim Interpretation

As this Court itself has stated:[18]

18. *McClain v. Ortmeyer,* 141 U.S. 419, 424 (1891).

> The object of the patent law in requiring the patentee to "particularly point out and distinctly claim the part, improvement, or combination which he claims as his invention or discovery" is not only to secure to him all to which he is entitled, but to apprise the public of what is still open to them. The claim is the measure of his right to relief, and, while the specification may be referred to to limit the claim, it can never be made available to expand it.

Section 131 of the patent statute,[19] on the other hand, requires the Patent and Trademark Office to examine only "the alleged new invention." The courts have long ago made clear that this can only mean *the invention as claimed.*[20] The two-step "point of novelty" approach to § 101 consideration violates this statutory mandate in that it calls for the patent examiner to ignore what the applicant has "alleged [to be his] new invention" and to substitute instead the examiner's own analysis of what is being claimed.

Closely akin to the mandate of 35 U.S.C. §§ 112, 131 is the fundamental rule that during the examination process limitations may not be read into or read out of a claim. The many cases[21] that comprise this body of

---

19. § 131. Examination of application.
The commissioner shall cause an examination to be made of the application and the alleged new invention; and if on such examination it appears that the applicant is entitled to a patent under the law, the Commissioner shall issue a patent therefor.
35 U.S.C. § 131 (1976).

20. *See, e.g., In re Moore,* 429 F.2d 985 (CCPA 1970).

21. *See, e.g., In re Priest,* 582 F.2d 33 (CCPA 1978); *In re Deutsch,* 553 F.2d 689 (CCPA 1977); *In re Kebrich,* 201 F.2d 951 (CCPA 1953); *In re Fischer,* 113 F.2d 492 (CCPA 1940); *In re Crowell,* 84 F.2d 206 (CCPA 1936); *In re Krichbaum,* 39 F.2d 280 (CCPA 1930).

law have long recognized that the rule applies to *all* pre-issuance questions involving claim interpretation, both in the tribunals of the PTO and in the courts reviewing PTO actions.

The reasons for this rule were well stated by the CCPA in *In re Prater,* 415 F.2d 1393, 1404-05 (CCPA 1969) (footnotes omitted):

> [T]his court has consistently taken the tack that claims yet unpatented are to be given the broadest reasonable interpretation consistent with the specification *during the examination of a patent application* since the applicant may then *amend* his claims, the thought being to reduce the possibility that, after the patent is granted, the claims may be interpreted as giving broader coverage than is justified. . . . We are not persuaded by any sound reason why, at any time before the patent is granted, an applicant should have limitations of the specifications read into a claim where no express statement of the limitation is included in the claim.

Thus, it is well settled that, in the examination process, what is to be examined is the claims, nothing more and nothing less. Any test for determining the presence or absence of statutory subject matter which requires reading limitations out of the claim in question simply does not comport with this fundamental rule.

C. Dictum in Parker v. Flook Cited in Support of the Two-Step Approach Cannot be Justified

The majority opinion in *Flook* was quite specific in reciting the limited holding of that case. "Very simply, our holding today is that a claim for an improved method of calculation, even when tied to a specific end use, is

unpatentable subject matter under § 101." 437 U.S. at 595n.18. However, in reaching that holding, the majority appeared to set forth an analysis which requires that any hitherto unknown "principle of nature" embodied in a claimed invention must, nonetheless, be "assumed" to be within the prior art. It is this dictum upon which the government implicitly premises its logic that anything in a claim which is or relates to "a scientific principle, idea, concept, formula, or phenomenon of nature" *cannot be considered at all* in a determination of patentability, and if everything else in the claim "is old, the claim does not recite statutory subject matter," *i.e.*, is not even eligible for consideration for patent protection under § 101. *See* Pet. Br. at 9-11.

We submit that the Court in *Flook* could not have intended that this broad dictum be indiscriminately applied to cases having fact situations extending well beyond those in *Flook*. Were a contrary course of action followed and such dictum taken literally, *all* inventions could be urged to be unpatentable since all inventions can be reduced to underlying laws or principles of nature which, once known, make the implementation of such laws or principles obvious. The Court could not possibly have intended such a result knowing that a patent it had sustained in recent years would have been manifestly unpatentable had the scientific truth on which it was based been taken as part of the prior art.[22] Surely, this cannot be the law.

---

22. We speak of the "Adams battery," the patentability of which was sustained in *U. S. v. Adams*, 383 U.S. 39 (1966). The invention was a simple combination of magnesium and cuprous chloride electrodes in water. Claim 1 of the patent in suit read:

> A battery comprising a liquid container, a magnesium electropositive electrode inside the container and having an exterior

It is important to understand that technology has reached today or soon will reach the point where *every* invention is *expressible* in terms of mathematical relationships, even where the specific principles upon which the invention is based are not fully understood. Physicists today express even the "uncertainty" principle in mathematical terms. At least in part, this capability is based upon the use of the computer as a scientific tool for analysis and expression. If a properly programmed computer is fed enough information about the nature of an invention—about how it works, what functions are performed, what results are obtained—it can express almost any invention in terms of mathematical relationships. It should be clear then that *no* standard by which an invention is to be judged, whether for patentability itself or even for eligibility for patent protection, can properly be built upon whether the claim, *i.e.*, the word definition of the invention, characterizes all or part of the invention in mathematical or abstract terms.

Further, one of the fundamental bases for the constitutional provision authorizing the creation of a patent system was the inducement of the disclosure of discoveries in the "useful arts." In enacting § 101, Congress expressed its conclusion that the categories of subject matter recited therein are of such character that it is socially

---

terminal, a fused cuprous chloride electronegative electrode, and a terminal connecting with said electronegative electrode. 383 U.S. at 42. The Court expressly found that each of the elements of the Adams battery was well known in the prior art. Patentability rested entirely on the discovery that combination of the three elements produced an unexpected electrochemical reaction which could be employed in storage batteries. If that electrochemical "principle of nature" had been assumed to be within the "prior art," the claimed invention would have been quite obvious and thus, according to *Flook*, non-statutory subject matter.

desirable to induce their disclosure and the commitment of capital to reduce them to commercial practice. Clearly the public policy considerations expressed in the patent statute are not based upon nor do they even consider the degree to which current scientific analysis permits an invention to be reduced ultimately to abstract mathematical expressions.

As an illustration of the problem here involved, consider one use of computer technology in the carburetor of today's automobile. In older cars, the carburetor contains a fixed mechanical circuit through which fuel and air are mixed and combustion gases exhausted. At different accelerations, speeds and operating temperatures, the ratio of fuel to air in this fixed mechanical circuit varies considerably from that which would produce optimum engine combustion. The result: inefficient power production, generation of excessive pollutants in the exhaust, and waste of fuel. In some new cars, however, manufacturers are installing microprocessors programmed to calculate optimum fuel/air ratios instantly in response to sensed changes in conditions and to cause modifications to take place within the carburetor to assure that carburetion is always at or near optimum. Reportedly these microprocessors enable an automobile to get up to 10% more mileage with up to 15% less pollutant output. Is this not the type of "useful art" for which the patent system was authorized by the Constitution and created by Congress? Of course, if the computer-control circuits and mechanisms would have been obvious to workers in the pertinent art, there can be no patent. But, clearly, the mere presence of programmed microprocessors should not dictate whether this improvement in the technology

of carburetors can be eligible for consideration for the patent grant.

Finally, we urge the Court to consider the chilling effect which the government's proposed "no-patent" rule would have on just one of the many incentives for which the patent system was designed — the incentive to disclose. Many innovations in the technological arts do not result from a "flash of genius," but rather are the product of painstaking trial and error.[23] Often, the scientific principles explaining the significant overall advance embodied in the end result cannot immediately be explained. Thus, it is a tenet of patent law[24] that the applicant for patent is not obliged to explain or even set forth the principles on which his invention is based. Yet, is it to be the law that the inventor who understands and ex-

---

23. Nevertheless, "[p]atentability shall not be negatived by the manner in which the invention was made." 35 U.S.C. § 103 (1976).

24. Consider, *e.g.*, this Court's statement in *Diamond Rubber Co. v. Consolidated Rubber Tire Co.*, 220 U.S. 428, 435 (1911):

> A patentee may be baldly empirical, seeing nothing beyond his experiments and the result; yet if he has added a new and valuable article to the world's utilities, he is entitled to the rank and protection of an inventor. And how can it take from his merit that he may not know all of the forces which he has brought into operation? It is certainly not necessary that he understand or be able to state the scientific principles underlying his invention, and it is immaterial whether he can stand a successful examination as to the speculative ideas involved. . . . He must, indeed, make such disclosure and description of his invention that it may be put into practice. In this he must be clear. He must not put forth a puzzle for invention or experiment to solve, but the description is sufficient if those skilled in the art can understand it. This satisfies the law, which only requires as a condition of its protection that the world be given something new and that the world be taught how to use it. It is no concern of the world whether the principles upon which the new construction acts be obvious or obscure, so that it inheres in the new construction.

plains the newly discovered (or appreciated) principle(s) embodied in his invention cannot obtain a valid patent *unless* he also claims a *nonobvious* application of the principle which is *nonobvious even with the principle known,* while the inventor who does not disclose or cannot explain the principle(s) underlying his invention has his evaluated solely against what the statute defines as prior art? Such a rule would inevitably result in the *non*-disclosure of many newly-discovered phenomena of nature — and this at a time when our society in the United States is deploring the slowdown in innovation and is seeking ways to stimulate its growth.[25] Yet such a rule would also follow from the "logic" of the two-step approach and, in fact, is already being applied by the PTO where that agency cannot hold, as it did here, that a patent claim, considered without that which is "non-statutory," defines only the "conventional." If a newly-discovered or appreciated phenomenon or principle of nature which is recited in a patent claim is not to be considered at all in the examination of that claim, then *the rest of the claim* must define that which is novel and not obvious in order for the claim to be held patentable.

It thus follows, according to the "logic" of the two-step approach, that the discoverer of a previously unknown or unappreciated principle, such as a mathematical relationship, must also discover or invent a non-obvious implementation of that principle in order to obtain a patent. Is this the legacy which the Court intended for *Parker v. Flook?* We do not believe so.

---

25. *See, e.g.,* SENATE COMM. ON COMMERCE, SCIENCE, AND TRANSPORTATION, 96th CONG., 1ST SESS., REPORT ON NATIONAL STRATEGY FOR TECHNOLOGICAL INNOVATION (Comm. Print 1979).

D. The Decisions in Benson and Flook Can Reasonably Be Urged to Be Consistent with a Rational Philosophy Found in Past Decisions of this Court Which Supports the Constitutional Goals of Our Patent System But *Does Not* Support the Artificial Analysis Proposed by the Government

The APLA interprets the decision in *Benson* to be an expression of the following policy regarding patent law: A patent claim drawn so generally as to preempt a mathematical algorithm in its entirety will be treated no differently than any other claim to a principle of nature. It will be held non-statutory, irrespective of the fact that the mathematical algorithm is intended to be implemented only by means of a computer, or that the only practical use for the algorithm is in association with a computer.

The following implication of the *Benson* holding eventually became clear: the fact of "computer implementation," whether it is "read into" a claim or expressly recited in the claim, will not remove from that claim the objection that it is drawn in general terms to an abstract principle and, as such, defines subject matter outside the patent statute. We accept this implication as being both pragmatic and consistent with other patent law. The digital computer is a tremendously useful technological tool. Its capacity to resolve complex mathematical relationships with incredible speed has led to the development of new mathematical relationships, previously unsolvable, to be used to accomplish useful technological goals. But this fact, that such a newly developed principle can only be implemented by computer, even coupled with the fact

that the relationship has value only in the environment of the practical end use contemplated, should not necessarily determine whether a claim which recites *only* the mathematical relationship or a mathematical manipulation thereof is eligible for patent protection.

In this Court's decision in *O'Reilly v. Morse,* 56 U.S. (15 How.) 62 (1853), the claim struck down was worded to preempt *all* practical applications of the disclosed new principle. The Court's conclusion that such claim would in effect preempt the principle itself, was an important factor in its decision holding that claim unpatentable. In *Benson,* patent claims which were drawn to *the only* practical application of the scientific principle involved, a mathematical relationship, were struck down because they also amounted in effect to a preemption of the principle itself.

Accordingly, the underlying philosophy of the *Benson* decision is consistent with the Court's view, going back more than one hundred years, that patents were not intended to be granted on general principles of nature, even those which had never before been discovered or appreciated. Rather, since the patent system was intended "[to] promote the Progress of . . . useful Arts," the discoverer of a principle of nature must first implement that principle in the practical, specific manner in one of the "useful arts" in order to be eligible for a patent. A claim which is no more specific than to recite the general application of the newly discovered principle will fail because it is an attempt to preempt more than is permissible. Similarly, a claim which fails to recite or make clear the specific implementation of the recited principle to a practical end use can reasonably be con-

sidered to be no more than a claim on the principle itself.

The decision in *Flook* can be viewed as an extension of the holding in *Benson* and consistent with its underlying philosophy. In *Flook,* the claims presented a mathematical relationship and solved it to produce a number. It was recited that this number was then to be used as an updated alarm limit in the control of petroleum refining processes. Thus, there was no apparent attempt to preempt all possible uses of the mathematical principle. Nevertheless, the majority in *Flook* found the claims to be non-statutory, holding that methods of calculation, even improved methods intended for a specific end use, should not be eligible for patent protection.

The *Flook* decision was not required by the holding or by the underlying philosophy in *Benson.* It was probably also not required by the Constitution. Nevertheless, the decision is consistent with *Benson* in that it gives no weight to the fact that the claimed mathematical relationship can only be practically implemented through the use of a computer. It is consistent also in its implication that a mathematical relationship is not the kind of discovery for which the patent system was intended. *Flook* extended this reasoning to hold, in effect, that the mere solution of a mathematical relationship, even in a technological environment, is not that application of the principle which converts it to a patentable "useful art."

However, the APLA sees nothing in *Benson,* in *Flook,* or in the two cases taken together, which requires that a claim be held ineligible for patent protection *merely because it includes a mathematical relationship, e.g.,* as a means of defining a structural relationship between

physical components of an otherwise "useful" device or apparatus, *or because it solves a mathematical relationship* as one step of a process leading to a specific, practical end use.

In *Flook,* the end result of the claimed method was simply the solution of a newly recognized mathematical relationship to produce a number, and the replacement of a pre-established number with the number newly produced. Even when tied to the specific end use environment recited in the claim preamble, the claim still amounted to little more than a method of calculating a new value for a number obtained according to a mathematical formula. The recited subsequent use of the calculated result was not so much considered *irrelevant* by the Court, as it was considered insignificant. The Court's holding that Flook's method amounted to nothing more than "an improved method of calculation" makes it clear that it was the *generality* of the subsequent *application* of the newly-discovered mathematical principle that concerned the majority.[26]

Thus, as we see it, the *decision* in *Flook* would require that post-solution activity include a specific, practical application of the calculated result to a specific benefit, as opposed to a general suggestion that the calculation may be (somehow) usefully applied in the environment

---

26. The majority clearly was influenced by the fact that:

> The patent application does not purport to explain how to select the appropriate margin of safety, the weighting factor, or any of the other variables. Nor does it purport to contain any disclosure relating to the chemical processes at work, the monitoring of process variables, or the means of setting off an alarm or adjusting an alarm system. All that it provides is a formula for computing an updated alarm limit.

Parker v. Flook, 437 U.S. at 586.

of recited end use. While the *Flook* majority clearly went too far in apparently requiring an independent "inventive concept" in the recited application of the principle,[27] its implicit requirement of a more specific recitation of an application providing practical benefit is consistent with prior cases[28] and not out of line with the ultimate aims of the patent system.

Accordingly, the *mere fact* that a claim includes or is based upon a non-statutory principle of nature, has to be recognized as *irrelevant* to the question of whether the entirety of the claimed subject matter comes within one of the categories of inventions eligible to be considered for patent. What is important, and critically relevant, is the degree to which the claim is drawn to a practical application of the principle to a specific beneficial result or effect in one of the useful, *i.e.*, technological, arts. If, in addition, the subject matter embodying such prac-

---

27. This is the premise of Section C., *supra*.

28. Note, in particular, the consistency of this approach with that used by this Court in deciding *Brenner v. Manson*, 383 U.S. 519 (1966). In the *Manson* case, the claimed invention was a process for making a chemical compound which, although heretofore known, had no known practical use. The compound was little more than a scientific curiosity. While recognizing that a process for making a chemical compound would ordinarily be the kind of useful process for which the patent system was intended, this Court concluded that the lack of a specific beneficial usefulness of the end product prevented the claimed method from having the kind of usefulness required by 35 U.S.C. § 101.

To be "useful" in the sense of the patent statute, therefore, the Court requires that a claimed invention must have a readily available, specific practical use. In the area of computer-implemented inventions, the Court now takes the position that a claim presenting a mathematical algorithm or other abstract scientific principle to be implemented by computer will not be considered eligible for patent protection unless that principle is presented in the context of a specific practical *application* which produces a result or effect having a readily ascertainable, commercial benefit in one of the technological arts.

tical application also satisfies the statutory *conditions* for patentability, the grant of limited exclusive rights is undeniably justified.

E. Conclusion

We have asked the Court to compare the "question" on which the government sought certiorari in this case with the actual language used in the claims of the application before the Court. If this question is presented by this case, it can only be through a process of analysis which *ignores* the clear language of the claims, *ignores* decades of customary procedure, and *ignores* a long legal history of repeated insistence by the CCPA on specific application of the statutory language of the patent code in making rejections of an applicant's claims for patent protection. If this question is presented, it can only be by a claim dissection process which uses the meat axe, not the probing scalpel—a process which destroys the basic framework of the claims and changes them beyond recognition.

Respondents here are not claiming "computer programs." Nor does this case involve any attempt to "word" a claim through use of the draftsman's skills to make it appear as if it is drawn to something else when it in fact covers nothing but a computer program. If these claims are considered in their entirety, it is clear that no "principle," or abstract proposition, will be preempted by any patent which may issue. Rather, what will be patented will be a very practical, very beneficial *application* of programmed data processors producing desirable results in the technological arts. The disclosure of this

type of invention is precisely what our patent system was intended to encourage.

As we have shown, the logic of the position being asserted by the government will result in a rule which could strike down many meritorious inventions and which could lead to the non-disclosure of previously unkown or unappreciated principles and phenomena of nature. This cannot be the intent of the Court.

The APLA urges the Court to recognize that

> labeling any invention as a "computer program" and, therefore, not eligible under § 101 to be considered for patent protection simply because the novel elements or steps of the invention reside in the use of a programmed computer, cannot be justified under the patent statute or prior decisions of this Court; and that

> claims involving computer programs can be analyzed according to the principles expressed in the existing case law from this Court and from the CCPA without having to rely on Congress to act.

The judgment below, in its holding that § 101 is not a proper basis for rejecting this application, should be affirmed. Additionally, or in the alternative, if the Court deems it appropriate, this case should be remanded to the PTO for further consideration, consistent with the

Court's opinion, of any remaining questions under §§ 102, 103 and 112 of the patent statute.

Respectfully submitted,

AMERICAN PATENT LAW
ASSOCIATION, INC.
2001 Jefferson Davis Highway
Arlington, Virginia 22202
DONALD R. DUNNER, President

KENNETH E. KUFFNER
ARNOLD, WHITE & DURKEE
2100 Transco Tower
Houston, Texas 77056
Attorney

Of Counsel:

TRAVIS GORDON WHITE
JOHN D. NORRIS
ARNOLD, WHITE & DURKEE
2100 Transco Tower
Houston, Texas 77056

Arlington, Virginia

July _____, 1980

Supreme Court, U. S.
FILED
AUG 4 1980
MICHAEL RODAK, JR., CLERK

Nos. 79-855 and 79-1112

IN THE

Supreme Court of the United States

October Term, 1979

No. 79-855.

SIDNEY A. DIAMOND, Commissioner of Patents and Trademarks, *Petitioner,*

v.

JOHN J. BRADLEY and BENJAMIN S. FRANKLIN, *Respondents.*

No. 79-1112.

SIDNEY A. DIAMOND, Commissioner of Patents and Trademarks, *Petitioner,*

v.

JAMES R. DIEHR, II and THEODORE A. LUTTON, *Respondents.*

On Writs of Certiorari to the United States Court of Customs and Patent Appeals.

BRIEF AMICI CURIAE FOR APPLIED DATA RESEARCH, INC. (ADR) and WHITLOW COMPUTER SYSTEMS INC. (WHITLOW).

MORTON C. JACOBS,
Suite 1420,
1420 Walnut Street,
Philadelphia, Pennsylvania 19102
Attorney for Amici ADR and Whitlow.

International Printing Co., 711 So. 50th St., Phila., Pa. 19143 — Tel. (215) 727-8711

QUESTIONS PRESENTED.

A. Questions of Law.

1a. Whether subject-matter patentability of a machine under 35 USC 101 can be different for different modes of machine construction.

b. Whether a computerized machine or industrial process that is patentable subject matter under 35 USC 101 when constructed with a hardware program (wired circuits) would also be patentable subject matter when constructed with a stored computer program (i.e., firmware or software).

2. Whether the Supreme Court is an appropriate body for resolving the complex economic and public policy issues underlying the Patent Office request that the Court exclude from subject-matter patentability a computerized machine or industrial process implemented by a stored computer program where the Patent Office has already been to the Congress, and has failed to get such exclusionary legislation.

B. Questions of Fact.

1. Whether a general-purpose computer containing a novel stored program (firmware or software) is converted by the stored program into a different machine from the general-purpose computer alone.

2. Whether those skilled in the art commonly construct machines alternatively by hardware programs completely or by a general-purpose computer with a stored program (firmware or software).

C. Question of Fact and Law.

Whether a computerized machine built with a stored program differs from the same machine built with a hardware program in any respect that would be material to its subject-matter patentability.

TABLE OF CONTENTS.

	Page
QUESTIONS PRESENTED	i
A. Questions of Law	i
B. Questions of Fact	i
C. Question of Fact and Law	i
INTEREST OF AMICI CURIAE	1
SUMMARY OF ARGUMENT	6
ARGUMENT	8
I. The Patent Office in Effect Calls for a New Rule of Law That Would Predicate Subject-Matter Patentability of a Computerized Machine on Its Mode of Construction; Machines Built With a Stored Program (Firmware or Software) Would Not Be Patentable, Even Though Those Built With Hardware Programs Are	8
II. Machines Built With Stored Programs; the Myth of the Non-Machine	10
A. What Is a Machine?	10
B. Why No Factual Record of Building Machines Using the Restructuring Principle of Stored Programs	13
C. The Myth of the Non-Machine Dispelled	14
III. These Computer-Program Appeals Are for Administrative Convenience, Not to Resolve a Dispute as to Law or Fact	17
CONCLUSION	22
APPENDIX A:	
Excerpts From Brief Amicus Curiae for the Association of Data Processing Service Organizations (ADAPSO)	A1
APPENDIX B:	
Letter Dated January 26, 1979 From Jacobs to Baruch	A21
Letter Dated April 30, 1979 From Jacobs to Baruch	A24
Letter Dated May 28, 1979 From Baruch to Jacobs	A29

TABLE OF CITATIONS.

Cases: Page

Burr v. Duryee, 68 U. S. 531 ..... 10, 11
Dann v. Chatfield, No. 76-1559, cert. dnd (1977) 8
Dann v. Johnston, 425 U. S. 219 (1976) 7, 8
Dann v. Johnston, No. 74-1033 ..... 12, 13, 14, 15
Dann v. Noll, No. 76-1558, cert. dnd. — U. S. — (1977) 8
Data Cash Systems, Inc. v. JS&A Group, Inc., et al., 480 F. Supp. 1063 (DC, Ill. 1979) 21
Diamond v. Chakrabarty, — U. S. — (June 1980) 17, 20
Diamond v. Sherwood, pet. for cert. pending 8
Gottschalk v. Benson, 409 U. S. 63 (1972) 8, 20
In re Johnson and Parrack, 589 F. 2d 1070 18
In re Johnston, 502 F. 2d 765 (1974) 5
Parker v. Flook, 437 U. S. 584 (1978) 7, 8, 20

Statutes:

35 USC:
- Generally 20
- 101 ..... i, 6, 9, 10
- 102 9
- 103 9
- 106 19
- 131 19
- 135 9

United States Constitution:

Article I, Section 8 21

TABLE OF CITATIONS (Continued).

Authorities:	Page
Computer Software Management U. S. Dept. of Commerce; NBS Special Publication 500-11, July 1977	4, 5
Hearings on H. R. 5924 Before Subcomm. No. 3 of Comm. on Judic., April 20, 1967	19
Minsky, M. L., Computation: Finite and Infinite Machines, 1967, p. 201, f.n.	9, 11
2 New Encyclopedia Brittanica: Macropaedia, 505	11
Robinson on Patents (1890), Vol. 1, p. 257, § 173	10, 11
Science, Am. Ass. Adv. Science, Vol. 195, No. 4283, 1977	3
Technology Review, M.I.T., Vol. 72, pp. 18-19, May 1970	15

Nos. 79-855 and 79-1112

IN THE
SUPREME COURT OF THE UNITED STATES

OCTOBER TERM, 1979

No. 79-855
SIDNEY A. DIAMOND, Commissioner of Patents and Trademarks, *Petitioner*,
v.
JOHN J. BRADLEY and BENJAMIN S. FRANKLIN, *Respondents*.

No. 79-1112
SIDNEY A. DIAMOND, Commissioner of Patents and Trademarks, *Petitioner*,
v.
JAMES R. DIEHR, II and THEODORE A. LUTTON *Respondents*.

ON WRITS OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS.

BRIEF AMICI CURIAE FOR APPLIED DATA RESEARCH, INC. (ADR) AND WHITLOW COMPUTER SYSTEMS INC. (WHITLOW)

Applied Data Research, Inc (ADR) and Whitlow Computer Systems Inc. file this brief *amici curiae* in opposition to Petitioner, together with letters of consent of the parties in accordance with the Rules of the Court.

INTEREST OF AMICI CURIAE.

ADR was incorporated in 1959 and since then has been a small business engaged in the computer field in the

marketing of various computer services, and proprietary software products. ADR presently markets more than a dozen software products, and has been awarded two software patents, one of which was the first patent issued for a software product on the market.[1] This first patent was issued to ADR's AUTOFLOW System, which has been one of the most successful standardized software products on the market. ADR also was granted a software patent for a control system on the complex monitoring of usage of telephone lines for the improvement of telephone service.[2]

Whitlow Computer Systems Inc. was incorporated in 1969, and since then has been a small business engaged in the computer field in developing, marketing and servicing of a proprietary software product known as SyncSort. This SyncSort product provides a computerized sorting process for which a patent was recently issued.[3] This innovative sorting process has overcome important machine obstacles to the efficient performance of sorting, which is a major part of most business data processing. SyncSort reduces the amount of computer memory that is required and the amount of computer operating time, and has been recognized by wide acceptance in the computer industry.

Though small companies, both ADR and WHITLOW have successfully competed with a giant hardware man-

1. U. S. Patent No. 3,533,086 of Martin A. Goetz, issued October 6, 1970 for "Automatic System for Constructing and Recording Display Charts". The other ADR patent is U. S. No. 3,380,029 of Martin A. Goetz, issued April 23, 1968 for "Sorting System". The issuance of this patent received considerable publicity as a so-called "first software patent".

2. U. S. Patent No. 4,066,843 of David Waks, et al., issued January 3, 1978 for "Telephone Circuit Monitoring System". ADR has sold the business of this product line.

3. U. S. Patent No. 4,210,961 of Duane L. Whitlow and Azra Sasson, issued July 1, 1980 for "Sorting System".

ufacturer such as IBM in the technical development and marketing of their software products.

ADR and WHITLOW have the interest of advising the Court as to the technological basis for the granting of software patents such as those they have received.[4] Software patents are patents on special-purpose machine systems that, because of technological and economic reasons, are constructed by means of stored programs (i.e., software) and a general-purpose computer.

As software manufacturers, *amici's* experience agrees with the views expressed by Dr. Ruth M. Davis, former Director of the Center for Computer Services and Technology of the National Bureau of Standards (in an address at the 1977 ADAPSO Management Conference):

> "Software gives us the capability of using computers creatively, flexibly and economically. It is a product that can be designed to benefit users innovatively and uniquely."

However, Dr. Davis also concluded (Science, Am. Ass. Adv. Science, Vol. 195, No. 4283, 1977):

> "The first 25-year history of software has not been marked with the successes, the advances, or the decreasing costs associated with computer equipment or hardware . . . software is the most unsafe, the least understood, and the most expensive component of total computer system costs. Software development costs are almost 10 per cent of total system costs. . . . The really useful and exciting advances in computing will probably only proceed at the same pace as advances in software engineering. And this will be distressingly so."

---

4. ADR and WHITLOW have no interest in and take no position with respect to the patentability of the particular invention of respondents' patent applications.

Dr. Davis testified before Congress on the "crucial computer problem" confronting our nation:

> "The crippling problems are lining up in the software rather than in the hardware field. In particular, they fall under what we call the application-independent problem category . . . these are the software problems which are common to many applications." (Testimony given before the Government Activity Subcommittee, House Committee on Government Operations, May 16, 1972).

This software is the type universally used, and it strongly affects machine efficiency; therefore, a hardware manufacturer may not always have the incentive to improve the software, where the "crippling problems have been". Dr. Davis has also testified to "The small, mostly 'young and struggling', companies who 'surely possess excellent technical innovative capabilities' and 'the principal technical capability . . . to work . . . on our critical computer problems" (Committee Testimony, *supra*, p. 15). The Assistant Secretary of Commerce for Science and Technology in 1979 summarized his agreement (see attached App. B, A29) as follows:

> "There is a need for software innovation in the computer industry and there is a small business potential to innovate in the software field. We agree too that the patent system is important in stimulating technological innovation."

The patent incentive is both a form of recognition and a major aid in obtaining the necessary financing without which these small companies face the prospect of shriveling and dying. The need of the incentive of patents for software is as least as great as that of the incentive available for hardware, because: "Today, providing computer software involves greater . . . risk than providing computer . . . hardware. . . ." (Computer Software Management.

U. S. Dept. of Commerce; NBS Special Publication 500-11, July 1977, p. 1).

Honeywell, the respondent herein, has divergent interests in patent protection for stored-programs, which take two forms, namely software and firmware. Honeywell is opposed to patents for software-built machines (see Br. Opp. 2).[5] In his dissenting opinion in *In re Johnston*, 502 F. 2d 765, 774 (1974), Rich, J. noted the economic interest of the hardware manufacturers in opposing such patents:

> ". . . the collective forces of major hardware (i.e., computer) manufacturers and their representative associations . . . for economic reasons, did not want patents granted on *programs* for their machines."

The Brief *amicus curiae* for National Semiconductor Corporation misstates (at p. 2) this interest of the hardware manufacturers who have consistently opposed patents for software-implemented inventions, as does National Semiconductor. Moreover, National Semiconductor's brief takes note of the "conversion to hardware of what was previously deemed software" (at p. 3), and indicates that its interests are to be free of patents for machine inventions (initially implemented in software), so that it can undertake "conversion to hardware" without restriction by a software inventor's patent.

The real strides forward in computer development have occurred in hardware, an area fully protected by the patent laws. They have not occurred in software, "the most expensive component of total computer system costs." (Dr. Davis, *supra*). It would seem evident that the profit-making corporations that oppose patents for the software technology do so, because it would be to their profit advantage.

---

5. Brief in Opposition of respondent Bradley; references to briefs are to those in the *Bradley* case, except where otherwise indicated.

SUMMARY OF ARGUMENT.

The writ of certiorari should be dismissed as improvidently granted. For the Court to resolve what the Commissioner of Patents presents as the key issues in this case, it would have to undertake a thorough inquiry into the complex technological facts of the construction of computerized machines. But the record does not provide the factual basis which would enable the Court to make such an inquiry, and to review the fact finding by the C. C. P. A. into the nature of this technology, which the C. C. P. A. resolved in prior cases. The Supreme Court is not the appropriate forum for initial fact finding and though eight computer program cases have been brought to the Court by the Patent Office on certiorari, none has contained a factual record of the nature of this technology.

Nominally the question presented by the Patent Office is whether computer programs are patentable subject matter under 35 USC 101. However, the real issue, not acknowledged by the Patent Office, is that by implication the Patent Office calls for a new rule that would predicate subject-matter patentability of a computerized machine on its mode of construction: While machines built with hardware programs are patentable, those built with stored programs (firmware or software) would not be patentable. There is no basis in law to deny patentability to a machine having one mode of construction as against another. That this issue of law is the underlying one in this case can be appreciated from a basic technological fact that a stored program restructures a general purpose computer into a different machine. That technological fact has been found by the C. C. P. A. and the Board of Appeals in predecessor cases. The underlying position of the Patent Office in this case is to ignore these predecessor findings of fact, and to assert that all computer pro-

grams are in the form of algorithms, and whether directed to mathematics or not, they are unpatentable under this Court's prior rulings in *Benson* and *Flook*. The C. C. P. A. and the Board have both found the basic technological fact; their finding should be given special weight by the Court, since each has special credentials of technological expertise, and has found in favor of the restructuring principle.

The background of these cases shows a history of the Patent Office seeking to avoid the administrative difficulties of the examination of computerized machines built with stored programs. The Patent Office unsuccessfully sought legislation from Congress to exclude such inventions from subject matter patentability. Thereafter, the then Commissioner of Patents announced a policy against the issuance of software patents to discourage the filing of such patent applications. The predecessor Commissioner decided not to pursue the practice of appeals to the Court, but with the change of Patent Office administration, the current cases have been brought. The stored-program technology antedates the Patent Code; it is not a new technology requiring special consideration of Congress.

The myth that computerized machines built with stored programs are non-machines is built on *legal* argument. The facts that a computerized device built with a stored program is a machine, and that the innovative aspect of it is a machine structure become plainly apparent when a few simple illustrations are considered. Copyrights would not protect such innovation.

ARGUMENT.

I. The Patent Office in Effect Calls for a New Rule of Law That Would Predicate Subject-Matter Patentability of a Computerized Machine on Its Mode of Construction; Machines Built With a Stored Program (Firmware or Software) Would Not be Patentable, Even Though Those Built With Hardware Programs Are.

The rule proposed by the Patent Office would in its effect establish subject-matter patentability based on the way a machine is built, a false dichotomy which has no support in the Constitution, in statute, nor in case law. This rule can be understood by an example; though hypothetical, it is based upon real experiences:

An inventor demonstrates his new invention to his patent attorney with great pride; he has developed a machine for reading books out loud to the blind. The machine is composed of both a reading and talking computer. After the demonstration, the patent attorney responds:

> "What's inside the cabinet? Did you build it with software or hardware (a stored program or hardware program)? If built with a hardware program, your machine would be patentable. But if you built it with a stored program, the Patent Office would say it was merely mathematics and, therefore, unpatentable."

That state of the law is what the Patent Office has sought in each of the eight "computer program" cases[6] that it has brought since 1971. But such a rule that would set up a dichotomy in subject-matter patentability based

6. *Gottschalk v. Benson*, 409 U. S. 63 (1972); *Dann v. Johnston*, 425 U. S. 219 (1976); *Parker v. Flook*, 437 U. S. 584 (1978); *Dann v. Chatfield*, No. 76-1559, cert. dnd (1977); *Dann v. Noll*, No. 76-1558, cert. dnd. — U. S. — (1977); and besides these two cases, that of *Diamond v. Sherwood*, pet. for cert. pending.

on the mode of construction would make no sense in fact or in law. The reading and talking computer has all the attributes of utility and patentable invention whether built with hardware or a stored program. The stored-program construction is often the more powerful mode of machine construction (Minsky, M. L., Computation: Finite and Infinite Machines, 1967, p. 201, f.n.); today it is probably the only practical way to build the bookreading machine, both from the standpoint of time and cost.

To deny subject-matter patentability to the stored-program embodiment would be contrary to all these statutory requirements: (i) the inventor shall be first (35 USC 102); (ii) priority shall be determined on reduction to practice, e.g., the first successful construction and operation (35 USC 135); and (iii) "Patentability shall not be negatived by the manner in which the invention was made" (35 USC 103).

The Patent Office briefs bypass the existence, let alone patentability, of hardware programmed machines and state the Questioned Presented as follows:

> "Whether a computer program . . . is patentable subject matter under 35 USC 101." (Br. P. 2; see also Petitioner's Brief in *Diehr*, at 2, and Petition in *Sherwood*, at 2).

The term "computer program" used by the Patent Office applies only to one form of "computer program", the stored program (firmware or software). The Patent Office does not acknowledge that the computer experts tell us that hardware and stored programs serve as alternative modes of construction (see Minsky, *supra*, and attached App. A, A12-17), nor does the Patent Office acknowledge that it considers computerized machines built with hardware programs to be patentable under 35 USC 101.

It would seem clear that if the Court is to adopt the rule requested by the Patent Office that machines built

with stored programs are not patentable (but those with hardware programs are) the Court would have to deal with the following questions:

a) In what ways does a computerized machine built with a stored program differ from the same machine built with a hardware program?

b) Are any of those differences between hardware-program and stored-program machines material to the issue of subject-matter patentability?

The factual record supplied by the Patent Office does not enable the Court to deal with either of those questions. The Patent Office has not even supplied the Court with the technological facts of stored-program machines though it seeks a sweeping rule in these cases excluding them from 35 USC 101.

For that reason alone, the Patent Office case is fatally flawed, and the writs of certiorari in *Bradley* and *Diehr* should be dismissed as improvidently granted.

II. Machines Built With Stored Programs; the Myth of the Non-Machine.

A. What is a Machine?

Though we are here dealing with a class of machines that were unimaginable in the 19th century it is useful to start with the definitions of that era:

As the Court said in *Burr v. Duryee*, 68 U. S. 531, 570:

"A machine is a concrete thing; consisting of parts, or of certain devices and combination of devices. The principle of a machine is properly defined to be . . . that peculiar combination of devices which distinguishes it from other machines."

Robinson on Patents (1890), Vol. 1, p. 257, § 173, defines a machine as:

> "An instrument composed of one or more of the mechanical powers, and capable, when set in motion, of producing, by its own operation, certain predetermined physical effects. It is an artificial rule of action, receiving crude mechanical force from the motive power, and . . . transforming . . . it according to the mode established by that rule."

A renowned computer scientist and Professor of Electrical Engineering at the Massachusetts Institute of Technology has given this definition:

> "When we talk about a machine we have in mind not only (1) an object of some sort, but (2) an idea of what that object is supposed to do. . . . One expects a machine to have a significant number of 'parts' and to perform some reasonably complex operation on something, but it is difficult to capture in the same definition machines which peel apples and machines which transform coded information signals!" (Minsky, *supra*, p. 3).

The program is the "rule of action" (Robinson *supra*) or "principle" (Burr, *supra*) of a computerized machine. It determines what the machine "is supposed to do," (Minsky, *supra*), "*what* the automated system shall do and how the parts of the system must function in order to accomplish the desired result." (2 New Encyclopedia Brittanica: Macropaedia, 505, 506; see App. A, A8, f.n. 38).

A fundamental technological feature of the computer technology is the use of programs to structure the desired machine: namely, the state and configuration of the computer's electronic circuitry and other apparatus can be built with hardware programs or alternatively with stored programs. This fact of two modes of circuit configuration for building a computerized machine was admitted by

the Patent Office (in *Dann v. Johnston*, No. 74-1033, see Brief for Petitioner at 14-15) as follows:

> "The particular operations to be performed by a computer are governed by the state ['the internal molecular or electronic condition of a part'] and configuration ['the physical location and arrangement of parts'] of its electronic circuitry and other apparatus. Such circuitry can be fully built into the computer through wires and other physical elements (i.e., as part of the 'hardware'), in which case, unless it is physically rewired, the computer will ordinarily serve only a specific purpose or application.
>
> Alternatively, the computer may be designed so that the state or configuration of its circuitry can readily be changed without rewiring or other visible physical changes. Such changes are accomplished by the use of different programs, . . . (e.g., punched cards, tapes, disks) . . ."

This admission sums up the essential difference between building the rule of action (i.e., "operations to be performed") into a computer: either "through wires . . . as part of the 'hardware' . . . [or] without rewiring or other visible physical changes . . . by the use of different [stored] programs . . ." Moreover, the "configuration of its circuitry" in each case determines the rule of action. Therefore, changing that configuration changes the machine, and the circuit "configuration . . . can be readily changed [by stored] programs."

This Patent Office admission to the Court of the technological facts of computers and stored programs contradicts the factual assertion made by the Patent Office herein (Br. P. 8-9):

> "Firmware does not convert the conventional computer . . . into a new . . . machine",

and (Br. P. 21):

> "Nor does respondents' microprogram somehow transform the old hardware into a new machine . . ."

These factual assertions by the Patent Office are not supported by any factual (nor legal) authority, and they are not presented to the Court for the Court's *finding of fact.* To the contrary, they are presented to the Court for its approval *as a legal conclusion,* notwithstanding that it is contradicted by the finding of fact of the Patent Office Board of Appeals in *Ex parte Johnston* which reversed the Examiner on precisely this issue (see decision quotation in attached App. A, A3, f.n. 24).

A stored (wireless) program serves to configure the computer circuits, just as a hardware (wired) program does; and it is similar to the configuration of mechanical parts in a mechanical machine. In each case, the configuration of devices supplies the rule of action. To remove the stored program from the general-purpose computer is to remove its unique rule of action as a machine. This would be the same as untying the wires interconnecting the circuits of a hardware-program computer, or disassembling the wheels, gears and levers of a mechanically programmed machine; and conversely the same.

B. Why No Factual Record of Building Machines Using the Restructuring Principle of Stored Programs.

In none of the eight "computer program" cases has the Patent Office met its responsibility as petitioner to supply a factual record of the complex technology of computers and computer programs. Respondents on the other hand have always relied on the fact finding of the C. C. P. A., which has an established expertise in technology. Instead of giving the Court the facts on what is largely (and, *amici* submit, now entirely) an issue of technology, the prestigious Patent Office, noted for its knowledge of advanced technology, relies on legal argument

and rhetoric. When the Patent Office argues law and merely denies and disparages the facts, but does not develop a factual and explanatory record, the lawyer's old adage suggests: the technological facts are against it!

C. The Myth of the Non-Machine Dispelled.

The restructuring principle of the modern computer, whose stored program configures and controls the computer's circuits, taxes the credulity of the uninitiated. How can what seems to be a "recording" of electrical signals (like a musical record) produce a new machine? Such a question was directed from the Court during the *Johnston* oral argument (see attached App. A, A1-15 for an answer based on legal and scientific authorities). In *Johnston,* there were findings of fact accepting the restructuring principle by both tribunals below, the PTO Board of Appeals, as well as the C. C. P. A. Therefore, the Patent Office was compelled to admit in its *Johnston* brief (*supra*), that, by means of a stored program in a computer, the "configuration of its circuitry can readily be changed without rewiring or other visible physical changes."

No such acknowledgment has been made here; to the contrary, the Patent Office denies the restructuring principle (Br. P., 21):

> "Nor does respondents' microprogram somehow transform the old hardware into a new machine. . . . Still less does the recording of the microprogram in the control unit turn the combination of the programmed unit and the conventional elements of the computer into a new machine. The only machine involved here is the familiar general purpose digital computer, unchanged by respondents in any of its hardware components . . ."

The Patent Office offers no legal or scientific authority nor factual explanation for that denial of the facts, nor justification for extending it to the following analogy (Br. P. 18, f.n. 27):

> "It is as if respondents claimed a new machine from the combination of an old player piano and a piano roll inscribed with a song they had created".

The spuriousness of this analogy is readily seen, for a computer program has been used to restructure a general-purpose computer (e.g., the same computer used in the book reading machine) to make (give it the rule of action of) a musical synthesizer. The latter then operates like a player piano for receiving punch cards (piano rolls) as inputs and playing them (for a photographic illustration, see Technology Review, M.I.T., Vol. 72, pp. 18-19, May 1970, or the Appendix of Brief for Respondent in *Dann v. Johnston*, No. 74-1033). But a piano roll does not restructure an old player piano; it remains nothing more than a player piano even with the piano roll.

The restructuring principle is also easily seen in other well known examples of the use of stored programs: to build a talking computer; or a computer for recognizing (reading) printed characters; or to combine those two with additional stored programs to build the aforementioned machine for reading books out loud (for other examples, see attached App. A, A11).

The Patent Office argues that all stored programs are "necessarily . . . mathematical" (Br. P. 14):

> ". . . because a digital computer is designed solely to perform mathematical computations, any program that it can implement must necessarily be mathematical."

However, its premise is wrong, because among the most unique functions of general-purpose computers are those

of "Conditional Transfer of Control" (see App. A, A6-7). By means of these control functions, stored programs attain extraordinary flexibility and power in the operations they can perform. The variety of machines which have been, and can be, built thereby must boggle the imagination, and they are not just mathematical machines.

The Patent Office seems to contend that all innovation, in the stored program field, for some years past and forevermore, is merely a matter of coming up with an abstract idea, drawing some flow chart diagrams and giving it to a non-innovative programmer to "translate" into suitable code (Br. P. 12). The Patent Office argues, in effect, that the general-purpose computer is so extraordinary that this process can be applied to any machine that one can imagine; one need only say "there ought to be machine that . . ." and the appropriate routine tasks can be performed to build it. Needless to say, the inventor of the reading and talking computer would not recognize his own experience in such a disparaging and unrealistic description (see App. A, A17-20).

If there were any substance to this Patent Office argument that we can leave such new developments to the non-innovative programmer to "translate" into a program, it would bring to an end the need for inventions and "presage the arrival of that period when human improvements must end", as the first Commissioner of Patents, Henry L. Ellsworth, in his 1843 report to Congress (PTO Off. Info. Serv.) stated:

> "The advancement of the arts, from year to year, taxes our credulity and seems to presage the arrival of that period when human improvements must end."

Needless to say the millenium has still not arrived, the need for software innovation is still with us, and the incentives of the patent system are still required.

In *Diamond v. Chakrabarty*, — U. S. — (June 1980), the Court upheld the patentability of a living micro-organism constructed with a novel program. Certainly, since the latter is patentable subject matter, and since machines built with hardware programs are also patentable, there is no reason to exclude, from subject matter patentability, programmed machines built with stored programs.

In the absence of a factual record, neither the present nor the predecessor computer-program cases could deal with the underlying technological issues: the non-machine as against the restructuring principle of stored programs. The contentions placed before the Court suggest a game based on the story of the blind men and the elephant: "Feel this (the tail), it's a rope"; "And this (the ear), it's a fan"; and so on. Instead of the technological facts of these machines, superficial aspects are highlighted (all programs are mathematical or abstract ideas; the material is not transformed; the parts still work the way they were intended) without bringing to light the total nature and complexity of the programmed computer.

This blind-man game would quickly terminate after a simple physical demonstration to the Court—that of inserting a series of different programs into a general purpose computer to produce a corresponding series of different machines.

III. These Computer-Program Appeals Are for Administrative Convenience, Not to Resolve a Dispute as to Law or Fact.

In April 1979, representatives of two software companies (amici herein) met with the Assistant Secretary of Commerce for Science and Technology, the Honorable Jordan J. Baruch, to discuss patent protection for software (see correspondence thereon in attached App. B, A21-29).

This meeting was requested by software people following an address by Mr. Baruch to the patent bar on the Administration's patent policy and our nation's need for technological innovation. Mr. Baruch's background as a computer scientist and his administrative relation to the Patent and Trademark Office placed him in a special position to deal with this matter. Commissioner of Patents Donald W. Banner and Patent Office Solicitor Joseph F. Nakamura also attended the meeting.

Assistant Secretary Baruch wrote in summary of the meeting (App. B, A29):

> "We indeed had a very useful exchange of ideas. As you point out, we can quickly 'stipulate' to several things. There is a need for software innovation in the computer industry and there is a small business potential to innovate in the software field. We agree too that the patent system is important in stimulating techonological innovation."

The principal issue of discussion was the technological nature of computer programs. On this point, Mr. Baruch noted that, from an engineering viewpoint, choosing between software or hardware for building a computer system was essentially a matter of evaluating the relative costs (App. B, A24), that is, a matter of tradeoffs in economics and other factors of relative advantage. This was an acknowledgment of the well known technological principle of the engineering equivalence of software and hardware (see App. A, A12-17).

Thereafter, in June 1979, the Commissioner of Patents announced in an address to a patent bar conference that he would not seek certiorari to this Court from the C. C. P. A.'s reversal of the Patent Office in the computer programming case of *In re Johnson and Parrack*, 589 F. 2d 1070. Instead the Commissioner announced he would go

forward with the preparation of guidelines for the examination of inventions implemented with stored programs:

> "Before leaving the subject of computers, you may have noticed that the PTO did not petition for a writ of certiorari in the Johnson and Parrack case. On the contrary, we are at the present time giving thought to the preparation of guidelines on the patentability of computer programs." (PTO Off. Info. Serv.).

However, with the entry of judgment by the C. C. P. A. in the *Bradley* case on July 5, 1979, and the appointment of a new Commissioner of Patents (following the resignation of Commissioner Banner), a new decision was made to seek certiorari in the *Bradley* and *Diehr* cases.

This same pattern in the history of this issue has repeated for the past fourteen years. In 1966, the Patent Office issued Guidelines for Examination of Computer Programs which recognized the patentability of machines built by stored programs (see App. A, A2-3, f.n. 22, 23). Thereafter, the Patent Office sought legislation proposing 35 USC 106 (H. R. 5924 and S. 1042, 90th Cong. 1st Sess., 1967), which would have excluded a "plan of action" from patentability. This legislation failed when the President's Science Adviser discredited it (because it sought to exclude software-built machines from patentability, while granting patents for the equivalent hardware programs) (see Hearings on H. R. 5924 Before Subcomm. No. 3 of Comm. on Judic., April 20, 1967, at p. 160).

In 1968, Revised Guidelines were issued to refuse patents for stored-program inventions, based upon "policy" (i.e., not on a basis of fact or law), and in contravention of the mandate of 35 USC 131. The Commissioner of Patents stated the purpose of this "policy":

> "The present Office policy of refusing patents in this [computer program] field has acted as a strong deterrent to the filing of applications". (Commissioner

> Brenner, "The Future of Computer Programs in the U. S. Patent Office", Law of Software, 1968, p. B-12).

However, the C. C. P. A. in its appellate review of this Patent Office "policy" decision has overturned that decision, and in a line of cases (which follow this Court's ruling on the non-patentability of mathematics in *Benson* and *Flook*) upheld the subject-matter patentability of certain software-implemented inventions.

Now, the Patent Office again seeks relief from its mandated task of examining patents on software implemented inventions. Confronted by the implications of Commissioner Banner's decision to follow this Court's and the C. C. P. A's rulings and to establish new guidelines for the examination of computer programs, and confronted with the administrative difficulties that such examination may entail, the Patent Office is here again engaged in its Sisyphean task of asking this Court for relief from that inconvenience.

However, the Patent Office can be relieved from its mandate only by the Congress which has set it. As the Court stated in *Diamond v. Chakrabarty*, — U. S. — (June 1980):

> "Our task, rather, is the narrow one of determining what Congress meant by the words it used in the statute; once that is done our powers are exhausted".

Unlike the *Chakrabarty* case, this stored-program technology has been in existence since the 1940's, prior to the 1952 codification of the patent statutes by Congress in 35 USC (see App. A, A14-15, f.n. 60). The stored-programmed machine is an old technology, not a new one, and after more than thirty years of being in the mainstream of the developing "useful arts" should be recognized under the Patent Laws.

The Patent Office suggests (Br. P. 24) copyright protection; but that would be inadequate and inappropriate to the task, and destructive of the constitutional purpose of the patent system as an incentive to machine innovation. There is no evidence of intent of Congress to take away from software inventors the benefits of the patent laws.

Copright protects particular expression, not machine structure. There is no requirement of innovation, only originality; therefore, a copyright would not protect the machine configuration that is at the heart of the stored-program invention. See *Data Cash Systems, Inc. v. JS&A Group, Inc., et al,* 480 F. Supp. 1063 (DC, Ill. 1979), where the court *found as a fact* that "the computer program [stored in a computer] is a mechanical device which is engaged in the computer to become an essential part of the mechanical process"; f.n. 4, ¶ 2; it is therefore, not coprightable subject matter.

The fact that a "program" can be represented as a list of computer operations is no more significant than the fact that a radio circuit can be represented in a schematic diagram. An innovative radio would not be protected by a copyright on a drawing and the firmware-built computer that is respondent's invention would not be protected by a copyright on a program listing. *Data Cash, supra.*

Legislation to amend the copyright laws (Br. P. 24), if it passes Congress, will ultimately be reviewed by this Court to determine whether the mechanical device of a stored program in a computer is a "writing" within Article I, Section 8 of the Constitution. That copyright decision of the Court some years hence, cannot affect the patent decision in these cases.

CONCLUSION.

For the reasons stated above, it is respectfully urged that the writs of certiorari be dismissed. In the alternative, if this Court reaches the merits of the case, the decision of the Court of Customs and Patent Appeals on patentable subject matter should be affirmed.

Respectfully submitted,

MORTON C. JACOBS,
Attorney for Amici ADR and WHITLOW.

APPENDIX A

IN THE

Supreme Court of the United States

OCTOBER TERM, 1977

No. 77-642

LUTRELLE F. PARKER,

ACTING COMMISSIONER OF PATENTS AND TRADEMARKS,
Petitioner,

v.

DALE R. FLOOK,

Respondent.

ON WRIT OF CERTIORARI TO THE UNITED STATES COURT OF CUSTOMS AND PATENT APPEALS

EXCERPTS FROM BRIEF AMICUS CURIAE FOR THE ASSOCIATION OF DATA PROCESSING SERVICE ORGANIZATIONS (ADAPSO)

* * *

(page 16)

II. Special-Purpose Computers Built by Programming a General-Purpose Computer Form Machines and Machine Processes, Hence Patentable Subject Matter.

A. The Technology of Computers and Computer Programs.

The technological facts of the nature of programmed computers are made clear by (a) the computer experts,

(page 17)

(b) the C. C. P. A. findings in other cases, (c) the Patent Office's own findings some years ago, and (d) the Board of Appeals' specific fact finding in the *Johnston Case.*[17] The computer experts explain the facts of the well-known "feature distinguishing the computer from other mechanisms . . . its restructurability. . . . The program restructures the computer," [18] The findings of fact as to this computer technology by the C. C. P. A. in a long line of cases up to the present [19] have held that a general-purpose computer is a "storehouse of parts":

> But once a program has been introduced, the general-purpose digital computer becomes a special-purpose digital computer (i.e., a specific electrical circuit. . . .).[20]

The special-purpose computer "is physically different from the machine without the program" and is patentable as a machine defined as a new combination of elements under 35 U. S. C. § 112.[21] The Patent Office in its Guidelines for Examination of Computer Programs (1966) [22] written in

---

17. See f.n. 24, *infra.*

18. Professor Ivan Flores, City University of New York "Computer Software," *Science & Technology,* No. 89, May, 1969, pp. 16, 17.

19. *In re Prater and Wei,* 415 F. 2d 1393 (1969); *In re Bernhart,* 417 F. 2d 1395 (1969); *In re Comstock,* 481 F. 2d 905 (1973); *In re Knowlton,* 481 F. 2d 1357 (1973).

20. *In re Prater, supra,* f.n. 29.

21. *In re Bernhart, supra,* at 1400, Judge Rich in the *Johnston Case,* 502 F. 2d 765, 773 reaffirmed "the validity of this principle [namely] a new program makes an old general purpose digital computer into a new and different machine." There has always been unanimity and no dissent in the court below on "the validity of this principle."

22. 829 O. G. 441, 442, by Order of Commissioner of Patents Brenner, August 9, 1966.

(page 18)

part by members of the Patent Office Board of Appeals,[23] recognized that the stored program converts that " 'warehouse' of unrelated parts" of the general-purpose computer into a "special-purpose machine" which may be patented as "a novel and useful machine combination due to the interrelation of parts . . . and the fact that portions of the complete machine take the form of a replaceable program is of no moment."

In the *Johnston Case*, the Patent Board of Appeals found:

> . . . that an unprogrammed, or a differently programmed, general-purpose computer is [not] the same machine as the one that would result from appellant's programming of the same or of a similar computer.[24]

The following discussion of the nature of the computer and stored-program technology is presented in order to assist the Court in understanding that respondent's preferred embodiment of a programmed computer for automating an oil refinery's alarm limit control is a "concrete embodiment in a specially devised apparatus," the Patent Office's contrary and unsupported assertion (Br. Pet. 14) notwithstanding.[25]

---

23. *See* Commissioner of Patents Brenner, "The Future of Computer Programs in the United States Patent Office," *Law of Software*, 1968, p. B-8.

24. Reported in *Dann v. Johnston*, No. 74-1033, Petition for Writ of Certiorari, 25A, at 28A.

25. Petitioner's attempt to have this Court determine by legal fiat and contrary to the facts, the nature of computers and computer programs brings to mind the declaration of the Court of the Special Inquisition that the sun rotated around the earth in the face of Galileo's evidence of a different structural relation of sun and earth; *see* J. Bronowski, *The Ascent of Man*, (1973) p. 213, *et seq.*

(page 19)

1. Respondent's Programmed Computer for Automating an Oil Refinery Is a Circuit Configuration.

Respondent's preferred embodiment of a programmed computer is constructed as a configuration of electrical components formed in "a specific electrical circuit".[26] This structure of respondent's machine (and of other machines implemented with a general-purpose digital computer and an internally-stored computer program) is illustrated in the accompanying diagram of the hierarchical structure of a special-purpose computer (see p. 20).

In this diagram, each level of the hierarchy,[27] supplies the building blocks out of which the circuits of the next level of the system[28] are built. The bottom level consists of electrical components from which the logic circuits of the next level are configured. The logic circuits are combined into the more complex building blocks —circuits called the "functional units"[29] of the general-purpose computer's processor.[30] These functional units, in turn, are combined into the special-purpose computer (such as respondent's) which receives input signals (for the values of the oil refinery's "process variables") and pro-

26. *In re Prater, supra,* f.n. 29.

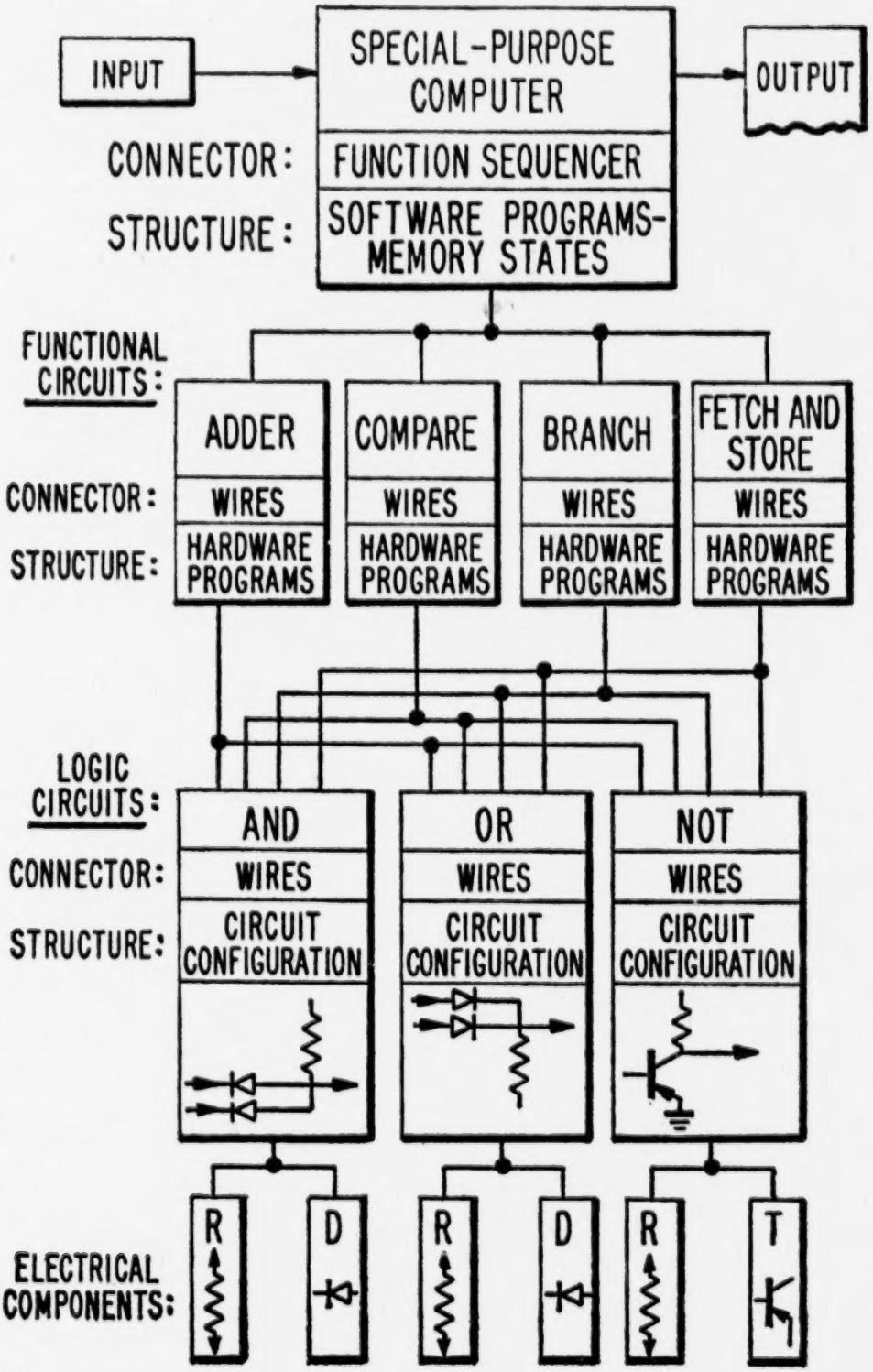
27. "*Hierarchy* . . . a rank arrangement in which the nature of function at each higher level becomes more broadly embracive than at the lower level." (A. Iberall, *Toward a General Science of Viable Systems,* McGraw-Hill, 1972, p. 388.)

28. "*System,* A complex unity formed of many, often diverse, parts subject to a common plan or serving a common purpose. . . ." (Ibid. at p. 7).

29. Professor Ivan Flores, *Computer Logic: The Functional Design of Digital Computers,* (W. L. Everitt, Editor) Prentice-Hall, Inc., 1960, Chapter Eleven, "Functional Units," pp. 173-207.

30. And into the other circuits (*e.g.,* memory, sequencer control, input and output) that form the general-purpose computer. "The term *logic* . . . describes the interrelation among the primary building blocks of the computer." Flores, *ibid.,* p. 123.

(page 20)


INPUT
SPECIAL-PURPOSE COMPUTER
OUTPUT
CONNECTOR:
FUNCTION SEQUENCER
STRUCTURE:
SOFTWARE PROGRAMS-MEMORY STATES
FUNCTIONAL CIRCUITS:
ADDER
COMPARE
BRANCH
FETCH AND STORE
CONNECTOR:
WIRES
WIRES
WIRES
WIRES
STRUCTURE:
HARDWARE PROGRAMS
HARDWARE PROGRAMS
HARDWARE PROGRAMS
HARDWARE PROGRAMS
LOGIC CIRCUITS:
AND
OR
NOT
CONNECTOR:
WIRES
WIRES
WIRES
STRUCTURE:
CIRCUIT CONFIGURATION
CIRCUIT CONFIGURATION
CIRCUIT CONFIGURATION
ELECTRICAL COMPONENTS:
R
D
R
D
R
T

(page 21)

duces output signals (the updated alarm values used to adjust the refinery's alarm limit).

The electrical components, e.g., resistors (R), diodes (D), transistors (T), at the bottom level [31] of the hierarchy are like those used in all modern electronic equipment (e.g., in radar, television, radio, telephone, automation). These electrical components are combined in different circuit configurations to form logic circuits called AND, OR and NOT which transform electrical signals in a manner representative of their named logic operation.[32]

The logic circuits are structured at the next level into functional circuits (Adder, Compare, Fetch and Store, and Branch or Conditional Transfer of Control). These functional circuits combine electrical signals in accordance with their named function. The Adder functions to add numbers represented by combinations of signals. Similarly, the Compare circuit operates to compare numbers for the relations of "equality" and "greater than" and "less than." Fetch and Store are control circuits for moving data signals to and from the memory, which is constructed from millions of flip-flop circuits, the basic storage units for the binary computer signals.[33]

The Branch or Conditional Transfer of Control performs a function unique to digital computers, whereby the sequence of functional circuits can be varied in accordance with different *conditions* of the input or processed

31. At a still lower level of this structural hierarchy, the molecular and atomic substructure of these components (such as the solid-state transistors and diodes) are important factors in their design and in their interrelationships at higher hierarchical levels.

32. The AND circuit produces an active output signal when all of the circuit's inputs are active; OR, when either one or another, or all of its inputs are active; NOT produces an active output when its input is not active. In place of three different types of logic circuits, many digital computers are built from combinations of a single such circuit, the NOR (the negative of OR; its output is active when neither of its inputs are active).

33. The flip-flop is also formed from the logic circuits.

(page 22)

data. In a simple example, different sequences of circuits are activated depending on which of two numbers is the greater.[34]

A general-purpose computer is configured to have a basic "repertoire" of functional circuits (also "instructions" or "orders").[35] The structural configuration of each of these functional, control and memory circuits is called "permanent programming"[36] or "hardware programming."[37]

Programs in the broader sense are part of all automatic machines, and automated systems. "*Program elements* determine *what* the automated system shall do and *how* the parts of the system must function in order to accomplish the desired result."[38]

"Software programming"[39] is the structure of the special-purpose computer which configures the functional circuits (Adder, Compare, Branch, etc.) to interconnect certain sequences of those circuits and combinations of those sequences.[40] The software programming (the in-

34. This Branch control (or "Decision") is fundamental to the ability to structure a general-purpose computer into a special-purpose computer by means of software programming.

35. A more elaborate set (*e.g.*, to perform also subtraction, multiplication and division, shifting of numbers, etc.) is often used.

36. I. Flores (1960), *supra*, p. 42.

37. Prof. Saul Rosen, "Hardware Design Reflecting Software Requirements," 33 *Proc. Fall Joint Computer Conference*, 1968, Pt. II, pp. 1443, 1444.

38. "The program may consist of mechanical parts such as gears, cams, and linkages that connect the various portions of a machine together and determine the timing of the various actions of the machine. In the most modern computer-controlled systems, the program may be stored in the form of punched paper cards, paper tape, magnetic tape, ferrite cores, electronic circuits, or other mechanical or electronic "memory" devices. 2 *New Encyclopedia Britannica: Macropaedia*, 505, 506, "Automation."

39. Term used herein for what is customarily called "computer programming" to avoid semantic confusion with hardware programming.

40. Basic repetitive sequences are called "sub-routines." (Br. Pet. 15).

(page 23)

ternally stored program) takes the physical form of combinations of different states of the memory circuits and of electrical signals that control the states and configurations of the functional circuits.

Software programming resembles hardware programming in a number of respects, and most importantly in the way that lower-level building blocks are structured.[41] However, the connector used with the software for configuring the functional units is a different type from that of the hardware programming. Starting at the bottom level of the hierarchy, as shown in the diagram, the electrical components are connected together in the logic circuits by fixed wire or "hardwire" connectors.[42] The logic circuits are similarly hardwire connected into the functional circuits and other circuits of the general-purpose computer. However, at the top of the hierarchy, a "softwire" connector is employed for the software structure. This connector includes a function ("instruction") sequencer, which effectively interconnects the functional circuits in the sequences prescribed by the software programming.[43]

By forming the structure at the top of the hierarchy as electrical signals, that structure can be stored in the computer memory in the same way that other signals are

41. The software and hardware programming are engineering equivalents, as discussed hereinafter.

42. "Hardwire" connections may be made by simply twisting or soldering wires together or in more advanced fabrication by printed circuits or integrated circuits; *see Scientific American* article, May, 1975, "Microprocessors."

43. The function sequencer activates by control signals the circuitry of the functional units at the appropriate times in sequences specified by the software programming. When activated the circuits are effectively "connected," and when inactive they are not. The engineer's term "softwire" refers to use of control signals for the connections. The sequencer, memory, input and output circuits are all part of the general-purpose computer; the control program is in the states of the memory circuits.

(page 24)

stored. The ability to store the structure of a special-purpose computer in the form of electrical signals gives rise to the ability to replace one such structure by another simply by replacing one set of electrical signals (software program) by another. *This concept of the stored-program computer is the foundation on which the computer industry is built.*

Since the building blocks of each hierarchical level of such computers are nested within those of the next higher level as shown in the diagram, the special-purpose computer at the top level includes all of the elements and structural relations shown in the drawing. This special-purpose computer receives input signals and, within its circuits, combines them and modifies them to produce output signals. The combining and modifying is in accordance with the structure of the stored software programming, which uniquely establishes the "rule of action" [44] of the special-purpose computer.

Respondent's special-purpose computer, in the same way, functions as "a specific electrical circuit" [45] configured out of sequences of functional circuits, which in turn are configured combinations of logic circuits, which in turn are configured combinations of circuit components.

2. The General-Purpose Computer Is an Incomplete Machine That Serves as "a Storehouse of Parts" and Is Completed by Respondent's Stored Program to Form a New Combination of Circuits as Respondent's Programmed Computer for Automating an Oil Refinery.

The general-purpose computer, though extremely complex, is an inchoate, and *incomplete* machine. The functional circuits (Adder, Compare, Branch, etc.) are

44. Robinson on Patents, Vol. I, p. 257, § 173.

45. *In re Prater, supra.*

(page 25)

not interrelated nor do they co-act in any useful way until structured [46] by the software programming into a special-purpose computer. The functional circuits are building blocks used by the software; and in that respect the general purpose computer is a " 'warehouse' of unrelated parts." [47]

This "storehouse of parts" [48] character of the general-purpose computer is evident from the hierarchical diagram on p. 21. Starting at the bottom level, each level serves as a storehouse of parts for the next higher level. The electrical components (resistors, diodes, transistors) serve no purpose by themselves and acquire functions only when connected in the various circuit configurations of the logic circuits. Similarly, the logic circuits serve only as a storehouse of parts for assembling the different circuit configurations of the functional units, and the latter circuits which constitute the general-purpose computer are interconnected by the software programming in the various sequences and combinations that make up the special-purpose computer.

The set of building blocks at each level is similar to the familiar toy Erector Set, whose parts can be connected in an unlimited number of ways to form machines of a wide variety of types. For example, with a storehouse of the electric components (resistors, diodes and transistors, etc.) almost all of the known electrical circuits can be built, including those of the computer logic circuits. With a storehouse of logic circuits, all of the various functional and other circuits of general-purpose computers can be built.[49] Similarly, the function circuits are interconnected

46. *See* Flores, (1969), *supra*, pp. 16, 17.

47. Patent Office Guidelines (1966), *supra*.

48. *In re Prater, supra*.

49. Special-purpose computers are also built directly from the logic circuits using permanent or hardware programming. One could also use the logic circuits as the functional units of the general-purpose

(page 26)

by the software into an endless variety of automatic machines—a musical synthesizer, the controls of a lunar landing module, a document reader for electric meter reports, a calculator for thereafter calculating and printing the associated electric bills. Respondent's programmed computer is similarly constructed in its preferred form of construction.

There is no predetermined relationship of the function circuits (Adder, Compare, Branch, etc. or their individual functions) to any of the above special-purpose computers and other devices; they are "*unrelated* parts" [50] (emphasis added). Nor is there a predetermined relationship of the resistors, diodes and transistors to a computer or television set. However, when the functional circuits of the general-purpose computer are structured by software programming—such as that of any of the above examples—these circuits are organized and interrelated by that programming just as are the electrical components in a television circuit. In that condition, the function sequencer makes the softwire connections called for by the stored program so that the functional circuits are connected into a special-purpose computer having the structure called for by the program.

The hardwire programming is used in building the functional circuits from the logic circuits and softwire programming is used in inter connecting the functional circuits into a special-purpose computer; both connect building blocks into more complex structures.

In short, as the Government asserted in another case, a software program may be "a technologically innovative

---

49. (Cont'd.)

computer, namely, the building blocks for software programming; but "such an approach would be impractically slow." Goldstine, H. H., *The Computer From Pascal to von Neumann*, (1972), p. 257.

50. Patent Office Guidelines (1966), *supra*.

(page 27)

product" [51] and may be demonstrated to have "technological superiority".[52]

The phrase "patents for computer programs" as used by the Patent Office to describe the key issue in these "computer programming" cases,[53] is technically imprecise and misleading in that what is sought to be protected in this technology is not some abstraction unrelated to a machine, but the structure of a particular special-purpose machine or machine-control process which may be produced by the softwired or programmed general-purpose computer but which also may be hardwired.

3. Hardware and Software Programming are Equivalent and Alternative Approaches for the Construction of Computerized Control Systems and Equally Patentable.

The Patent Office has set up a false dichotomy in the patentability of computerized control systems. Those implemented by hardware programming (*e.g.*, the general-purpose digital computer) are recognized to be machines by petitioner and patents covering such machines and their control processes are granted subject to their meeting the statutory tests of novelty (35 U. S. C. § 102) and unobviousness (35 U. S. C. § 103). However, computers implemented by software programming are generally denied patentability.[54]

---

51. Government's Pretrial Brief in *United States v. IBM*, 69 Civ. 200, at p. 202.

52. *Ibid.* at p. 29. This Government Brief in the IBM case makes it clear that the software industry is in the business of selling technological products (machines) not "abstract ideas."

53. *Benson, Johnston, Chatfield,* and *Noll,* the predecessor cases.

54. The Patent Office is not consistent in this respect; many of the Examiners understand the machine structure of software, as shown by the patent for an "Analytic Character Recognition System" implemented by software, U. S. 3,873,972.

(page 28)

Petitioner's distinction between the two alternative forms of computer construction is inconsistent with the well-established technological "principle of the equivalence of [software] programming and hardware. This principle [is based on the fact] that a program for a general-purpose machine turns it into a special-purpose machine." [55]

Petitioner has acknowledged in the previous *Johnston Case* this equivalence of hardware and software, namely, that a special-purpose computer can be built either by hard-wired logic circuitry or by programming a general-pupose computer.[56] The particular means that is used depends upon many business and engineering trade-offs.[57]

---

55. "The principle becomes more tangible when we compare in detail the logical design of that special-purpose machine with the flow chart of that program." Prof. Saul Gorn, Final Report No. AD 59 URI, at p. 4, from Univ. of Penna. Moore School of Elec. Eng. to U. S. Army Signal Research & Development Labs, 1959.

56. "The particular operations to be performed by a computer are governed by the state and configuration of its electronic circuitry [which] can be fully built into the computer through wires. . . . Alternatively, the computer may be designed so that the state or configuration of its circuitry can readily be changed without rewiring or other visible physical changes . . . by use of different 'programs.'" (See *Dann v. Johnston* No. 74-1033, Brief for the Petitioner at 14-15.)

57. "Perhaps the most important part of the job of computer system design is to determine which functions are best handled by hardware, and which should be provided by software. All of the traditional engineering considerations come into play here. One must consider the added cost of additional hardware measured against possible gains in operating efficiency. One must consider the availability of manpower with the appropriate skills for design and for implementation. One must also consider intangible factors such as the inherent advantages of simplicity and the hidden costs of complexity.

"From the point of view of the computer user it makes no difference which features are incorporated into the hardware and which are programmed in software.

"It is the combined hardware-software system, the 'extended machine' that the user must rely on to provide the power and flexibility and convenience that have been promised with each new generation of computing equipment." S. Rosen (1968), *supra*.

(page 29)

None of the factors concerning the choice of hard-or-softwiring has any relation to inventiveness of the underlying machine. If the circuit states and configuration of a particular special-purpose computer contain the elements of invention, the machine is patentable and the fact that the wiring is "hard" or "soft" is immaterial, for the two types of computer construction are interchangeable.

This equivalence is inherent in the concept of the stored-program computer. The first electronic computer, the ENIAC built about 1945, was "not a stored-program computer. Programs were installed and changed by engineers who changed the wiring among its various components."[58] The stored-program concept, invented while the ENIAC was being built, had as its very purpose the replacement of the hardwire connections by the softwire. The first published description[59] of the stored-program computer clearly explained (p. 1, ¶ 1.2) that the program ("instructions"[60]) was "an integral part of the [computer's] structure."

---

58. S. Rosen, Purdue University, "Electronic Computers: A Historical Survey," *Computing Surveys*, Vol. 1 (1969), pp. 7, 8.

59. A. W. Burks, H. H. Goldstine, J. von Neumann, *Preliminary Discussion of the Logical Design of an Electronic Computing Instrument*, 1946. In this famous "paper that became a landmark in the history of computer sciences if not in the history of human thought," (4 *New Encyclopedia Britannica: Macropaedia*, 1045, 1047) John von Neumann and his colleagues at the Institute for Advanced Study in Princeton first published "this stored-program concept" on which the modern computer is based and the technology of which underlies the issues in this case.

60. A human receives *instructions* on how to operate the manual choke in an automobile. When an *automatic* choke is built in by means of a "program," we don't think of "instructing" the engine or its controls. Yet that is the terminology of the computer industry established thirty years or more ago to describe the automatic controls of a stored program. The automatic control of the choke in future cars will be done by a more advanced programmed general-purpose computer, and appropriate computer "instructions" will be used for that purpose. *See* Vacroux, *infra*, f.n. 76, at p. 40.

(page 30)

As far back as 1949, when the first stored-program computer was constructed in England, its builders recognized the equivalence of programming by hardwire and softwire, and explained that additional functional circuits could be added to the computer or the equivalent software programming could be used to implement the same functions.[61]

The Patent Office has no difficulty in finding patentable subject matter in the hardware approach to building special-purpose computers which technologically began to be replaced 30 years ago with stored-program inventions. However, it insists that the stored program approach which is the present foundation of the computer industry and is revolutionizing the entire electronics industry is not patentable. To base disparate legal treatment upon a mode of construction has no justification in law, technology or common sense. Softwired (*i.e.*, wireless) programming is an historic advance over the wired program. The patent statutes and their constitutional base were not intended to be limited to a finite period of technological development. There is no justification for the Patent Office's position that technology has now outstripped the boundaries of patent law.

The equivalence of hardware and software underlies this revolutionary change taking place today in the electronics industry. Programmed microcomputers (general-purpose computers constructed with "microscopic components"[62]) are used to build electronic apparatus in "a wide range of applications from specialized laboratory instruments to sales terminals and electronic games."[63]

---

61. M. V. Wilkes, "Progress in High-Speed Calculating Machine Design," *Nature*, Vol. 164 (1949) pp. 341-3.

62. A. G. Vacroux, "Microcomputers," *supra*, pp. 32, 33.

63. *Ibid.*, p. 34.

(page 31)

"Until recently digital circuits were designed almost exclusively with 'hardwired' logic . . . although this approach is very powerful (indeed, it is used in most of the electronic equipment manufactured today), it yields an *inflexible* product that is difficult to modify or improve without major redesign. Most of the microcomputers in service today were adopted as a replacement for *inflexible* circuitry. . . . *it is to their flexibility, inherent in all stored-program systems, that they owe most of their popularity.*[64]

Thus, the general-purpose computer has become (in miniature size) a tool that is structured by the addition of programs to build all kinds of instruments and apparatus. Each progammed microcomputer is a different electronic machine as is each hardwired device that the programmed microcomputer is replacing. Such electronic machines have unquestionably been patentable subject matter under 35 U. S. C. § 101 when hardware programming was used. The objection to patentability based upon the use of the more modern and "flexible" technology of the stored program cannot be justified by any basis in law or fact.[65]

The denial of patents to a modern technology (the softwire mode of building special-purpose computers and other machines) raises anew the question: whether the

64. *Ibid.*, p. 39.

65. The failure to recognize the equivalence of hardware and software opens a pandora's box and leads inevitably to a "raiding" situation in which existing hardware patents have no legal protection from infringing software-constructed devices. *See Digitronics Corp. v. The New York Racing Association*, (E. D. N. Y.), 187 U. S. P. Q. 602, affd. on other grounds 553 F. 2d 740, cert. dnd., — U. S. —. This problem is not limited to the computer field, for many automatic machines have the potential to employ a programmed general-purpose computer to replace their existing electronic, electrical or mechanical controls. *See* 2 *New Encyclopedia Britannica, "Automation," supra.*

(page 32)

Patent Office "freeze[s] . . . patents to old technologies, leaving no room for the revelations of the new, onrushing technology." (*Benson, supra,* at 71.)

4. Software Is Not an "Abstract Idea" and Its Construction Is Engineering.

The software products segment of the computer industry is composed of hundreds of small and medium size companies, many of which are members of ADAPSO. The programmers build software out of building blocks to engineer the special-purpose computer in its hierarchical structure,[66] much the same as that required for other modern complex machines.

In the hierarchy of computer systems, the logic circuit designer builds logic circuits such as "AND" and "OR" circuits without concern for the molecular structure of the individual transistors and other components that make up those electronic circuits (which concern is left to component designers working at a lower level of hierarchy). The designer of the hardware function circuits such as "ADD," "COMPARE" and "STORE," configures logic circuits without concern for the detailed structure of the individual logic circuits he uses to construct the hardware function circuits. Similarly, the "programmer," the designer of the special-purpose computer, configures hardware function circuits into the structure of the ultimate special-purpose machine without concern for the detailed structure of the individual hardware function circuits. The same hierarchical process is followed if, instead of softwire programming, hardwire programming is used to build the special-purpose computer.

66. See "Hierarchies–Software Engineering Design," Software Engineering, International Computer State of the Art Report, 1972, p. 47 et seq.

(page 33)

These building blocks lie on successive levels of a hierarchy of circuit construction and the software engineering is performed at the highest level. The software itself, because of its complexity, often must also be constructed in a number of hierarchial levels, and the software engineer should preferably work like:

> ". . . the electrical engineer designing a new T.V. set [who] certainly doesn't design each vacuum tube, transistor, and capacitor all over again; he relies on existing components. In fact, he normally designs on a much higher level, using 'off-the-shelf' power supplies, oscillators, etc." [67]

A recent IBM report further demonstrates the similarities in software and hardware engineering:

> Structured programming reduces the arrangement of the program logic to a process like that found in [hardware] engineering where logic circuits are constructed from a basic set of figures. As such, it represents a standard based on a solid theoretical foundation.[68]

Working at different levels in the system hierarchy of the special-purpose computer to be built, contemporary software engineers as that IBM report explains, may use a "structured programming" [69] approach similar to the design of other automatic machines:

> The top-down approach is patterned after the natural approach to system design and requires that

67. The IBM report, "Composite Design: The Design of Modular Programs," Jan. 1973, p. 6.

68. "Management Concepts for Top-Down Structured Programming," IBM Report, February 1973, p. 1.

69. *Ibid.*, Abstract.

(page 34)

> programming proceed from developing the control architecture (interface) statements and initial data definitions downward to developing and integrating the functional units.

In the Apollo Space Shuttle Software System, a mammoth software development effort:

> . . . program structuring is of major importance; decisions . . . determine the total structure of the software both statically and dynamically. Program structuring defines the building blocks, the control mechanisms and the interfaces of the software. It is in this effort that the interrelationship of systems software, applications software and data is entirely mapped out.[70]

The software engineer builds complex automatic machines as the foregoing examples amply demonstrate. Neither the software engineer nor the hardware engineer devises "abstract ideas" or "mathematical algorithms," contrary to the Patent Office's suggestions (Br. Pet. 17) and to what CBEMA would have the Court believe (Br. CBEMA 17).

The design and development process in this technology is quite similar to that of any other technology. First the goal is decided upon, then the prior art is explored for possible solutions and, if not found, the "technological roadblocks" are identified.[71] Invention may take place in overcoming these roadblocks.

Parallels to the design and development process of other machines may be seen in the software design process

70. "Top-Down, Bottom-Up, Structured Programming & Program Structure," Draper Labs, MIT, Dec. 1972, pp. 28, 31.

71. *See, e.g., Computers in the 1980's,* R. Turn, Columbia Univ. Press, 1974, p. 21.

(page 35)

presented in a paper at the World NATO Conference entitled "Software Engineering" [72] which took place in Germany in 1969 and was attended by over 50 experts and prominent leaders in the software field.

° ° °

---

72. "Software Engineering," Report on a World Conference Sponsored by the NATO Science Committee, Garmisch, Germany, Jan., 1969, p. 13.

APPENDIX B.

LAW OFFICES
MILLMAN AND JACOBS
PATENT, TRADE MARK AND COPYRIGHT CAUSES
1420 WALNUT STREET, SUITE 1420
PHILADELPHIA, PENNSYLVANIA 19102

MORTON C. JACOBS
MAX R. MILLMAN

PHILADELPHIA (215) KI 6-5950
NEW JERSEY (609) 964-5588

January 26, 1979

The Honorable Jordan J. Baruch
Assistant Secretary of Commerce
for Science & Technology
U. S. Department of Commerce
Washington, D. C. 20230

Dear Dr. Baruch:

Your talk last week before the Philadelphia Patent Law Association on the role of the patent system in meeting our nation's need for technological innovation was most encouraging.

I was pleased to chat with you and to receive your favorable response to our request for a meeting with ADAPSO, the software association, which we represent as patent counsel and which is much concerned with stimulating innovation in computer technology and particularly in computer software.

The ADAPSO Software Protection Committee, in requesting a meeting with you, seeks to discuss how to facilitate for software innovators their rightful access to the incentives of the patent system.

ADAPSO's position is that patents should be granted for software-embodied inventions equally with those granted for hardware-embodied inventions. A choice between these two modes of its constuction should not make any difference in the patentability of an invention.

ADAPSO's view is based on the technological principle of the engineering equivalence of software and hardware—a principle which you readily acknowledged in response to a question after your talk.

To Hon. Jordan J. Baruch Page 2

January 26, 1979

In view of your grasp of the underlying technology, your direct participation is needed in answering this momentous and long-standing question, which the U. S. Supreme Court did not deal with in three "computer programming" decisions.

We would like to suggest a date for the meeting in March or April 1979 convenient to your calendar.

At your direction, we are concurrently channeling this request to the Honorable Joseph F. Nakamura, Patent Office Solicitor, who, you suggested, would participate. We concur (having worked with him for tens of years) in the high regard that you expressed for Mr. Nakamura.

A copy is also directed to the Commissioner of Patents for his information.

Sincerely yours,
Morton C. Jacobs

MCJ:swh

Copy: Honorable Donald W. Banner
Commissioner of Patents and Trademarks

Honorable Joseph F. Nakamura
Solicitor, U. S. Patent and Trademark Office

Mr. Bruce T. Coleman, Vice-President, ADAPSO;
Group Vice-President Software Products
Informatics, Inc.
Woodland Hills, California

Mr. Martin A. Goetz, Chairman
ADAPSO Software Protection Committee;
Senior Vice-President, Applied Data Research, Inc.
Princeton, New Jersey

Mr. Jerome Dreyer, Executive Director, ADAPSO
Arlington, Virginia

Milton R. Wessel, Esq., ADAPSO Counsel
Pleasantville, New York

LAW OFFICES
MILLMAN AND JACOBS
PATENT, TRADE MARK AND COPYRIGHT CAUSES
1420 WALNUT STREET, SUITE 1420
PHILADELPHIA, PENNSYLVANIA 19102

MORTON C. JACOBS
MAX R. MILLMAN

PHILADELPHIA (215) KI 6-5950
NEW JERSEY (609) 964-5588

April 30, 1979

The Honorable Jordan J. Baruch
Assistant Secretary of Commerce
for Science & Technology
U. S. Department of Commerce
Washington, D. C. 20230

Dear Dr. Baruch:

We are pleased with the meeting that ADAPSO held with you on April 3, 1979 to discuss patent protection for software. We were also pleased to meet at the same time with the Honorable Donald W. Banner, Commissioner of Patents and Trademarks, and the Honorable Joseph F. Nakamura, Solicitor, U. S. Patent and Trademark Office.

As you suggested, we can quickly "stipulate" that there is a need for software innovation in the computer industry and that there is a small business potential to innovate in the software field. There was also agreement on the importance of the Patent System to stimulate technological innovation.

We keenly appreciate that your expertise in this software technology enabled us to get to the essential issues. As you noted, the decision to build a machine in hardware or software is one based on engineering factors such as relative cost. Thus, these two forms of construction (known also as "hardwire" or "softwire") are engineering equivalents, a principle that has been understood from the beginnings of this technology some thirty years ago.

For that reason, and since the mode of constructing an invention (e.g., be it hardwired or softwired) can make no difference in its patentability, ADAPSO has for ten years and more been urging that the patent protection that has always been available for hardware-implemented inventions should likewise be available for the same invention that is software-implemented.

This should be plainly evident from technological considerations, for the form of an invention that is first built (for example, in the computer field) is frequently softwired. In patent law, the first form of an invention is also particularly significant, and often favored, for a variety of reasons.

The Supreme Court's decision in the five so-called "computer programming" cases that have been appealed reached no findings about the software technology that would suggest a different treatment for a software-embodied invention than for a hardware embodiment.

The Supreme Court's decisions that mathematics as such is not patentable are not contested by ADAPSO. It is our understanding that the Patent Office does not take the position that because an invention is describable in mathematical terms it is unpatentable. Though hardwired computer systems can be described by Boolean algebra, the Patent Office nevertheless issues patents for such systems.

The Court of Customs and Patent Appeals has felt qualified to deal with the technological nature of software from the very first case that came before it in 1969. In

that case, *In re Prater,* the Court concluded than an invention built with software would be patentable subject matter by the nature of the technology:

> . . . a general-purpose digital computer may be regarded as but a storeroom of parts and/or electrical components. But once a program has been introduced, the general-purpose digital computer becomes a special-purpose digital computer (i.e., a specific electrical circuit with or without electro-mechanical components) which, along with the process by which it operates, may be patented subject, of course, to the requirements of novelty, utility and non-obviousness.

The Patent Appeals Court has reaffirmed this technological principle numerous times over the years. Most

recently, in a December, 1978 decision *In re Johnson,* et al, the Patent Appeals Court continues to reject the false dichotomy between software-built and hardware-built inventions:

> We stated in In re Chatfield . . . that "the mere labeling of an invention as 'a computer program' does not aid in decision making" and the Supreme Court declined to decide either Benson or Flook on such broad, nonsubstantive grounds. . . .
>
> Very simply, our decision today recognizes that modern technology has fostered a class of inventions which are most accurately described as computer-implemented processes. Such processes are encompassed within 35 USC 101 under the same principles

> as other machine-implemented processes, subject to judicially determined exceptions, inter alia, mathematical formulas, methods of calculation, and mere ideas. The overbroad analysis of the PTO errs in failing to differentiate between a computer program, i.e., sets of instructions within a computer, and computer-implemented processes wherein a computer or other automated machine performs . . . the process. . . . This distinction must not be overlooked because there is no reason for treating a computer differently from any other apparatus employed to perform a . . . process. . . .

The Patent Office is not bound by an administrative decision taken ten years ago to rule out patents for software-built inventions, particularly where it is based on a technologically unsound premise, which the Patent Appeals Court has specifically rejected and the Supreme Court has declined to deal with. No change in the law is required, neither in the courts nor in Congress (the attempt ten years ago to have Congress legislate this rule of no-patents-for-software did not succeed).

To Hon. Jordan J. Baruch April 30, 1979

ADAPSO believes that the change that must be made is for the Patent Office to stop denying the technological principle of the engineering equivalence of hardware and software—to stop denying that software and hardware are but two different modes of machine construction. This error in how technology is viewed can be readily corrected.

Once that change is made, so that computer inventions are not treated differently as to patentability because of the mode of construction, the incentives of the Patent System can operate to stimulate innovation in software technology.

Sincerely yours,

MORTON C. JACOBS
Morton C. Jacobs
ADAPSO Patent Counsel

MCJ:swh

Copies: Honorable Donald W. Banner
Commissioner of Patents and Trademarks

Honorable Joseph F. Nakamura
Solicitor, U. S. Patent and Trademark Office

Mr. Bruct T. Coleman, Vice-President, ADAPSO
Group Vice-President Software Products
Informatics, Inc.
Woodland Hills, California

Mr. Martin A. Goetz, Chairman
ADAPSO Software Protection Committee;
Senior Vice-President, Applied Data
Research, Inc.
Princeton, New Jersey

Mr. Jerome Dreyer, Executive Director,
ADAPSO
Arlington, Virginia

Mr. Aso Tavitian, President
Whitlow Computer Services, Inc.

Milton R. Wessel, Esq., ADAPSO Counsel
Pleasantville, New York

(Letterhead of)

UNITED STATES DEPARTMENT OF COMMERCE
THE ASSISTANT SECRETARY OF SCIENCE AND TECHNOLOGY
Washington, D. C. 20230

(202) 377-3111 5/28/79

MAY 25 1979

Mr. Morton C. Jacobs
1420 Walnut Street
Suite 1420
Philadelphia, Pennsylvania 19102

Dear Mr. Jacobs:

Thank you for your letter of April 30, 1979 following up on our meeting of April 3 at which we discussed patent protection for software.

We indeed had a very useful exchange of ideas. As you point out, we can quickly "stipulate" to several things. There is a need for software innovation in the computer industry and there is a small business potential to innovate in the software field. We agree too that the patent system is important in stimulating technological innovation.

I very much appreciate ADAPSO's efforts to help clarify the issues with respect to this topic which has such important economic implications.

Sincerely,

JORDAN J. BARUCH
Jordan J. Baruch

Supreme Court, U.S.
FILED
JUL 9 1980
MICHAEL RODAK, JR., CLERK

No. 79-855

IN THE

Supreme Court of the United States

OCTOBER TERM, 1979

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS, *Petitioner,*

v.

JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN, *Respondents.*

On Writ Of Certiorari To The United States Court Of Customs And Patent Appeals

---

BRIEF AMICUS CURIAE FOR HALLIBURTON SERVICES

WILLIAM JAMES BEARD

2135 Highway 6
P.O. Box 42800
Houston, Texas 77042

Attorney for Amicus Curiae, Halliburton Services

Of Counsel;

JOHN H. TREGONING

Patent Counsel
Halliburton Services
P.O. Drawer 1431
Duncan, Oklahoma 73536

PRESS OF BYRON S. ADAMS PRINTING, INC., WASHINGTON, D.C.

TABLE OF CONTENTS

PAGE

I. The Nature and Interest of the Amicus 1

II. Summary of the Argument 2

III. Argument 3

A. The Patent and Trademark Office quest for a *per se* rule of unpatentability for any computer implemented process, however claimed, is not correct and not in the interest of justice. 3

B. The Court of Customs and Patent Appeals decisions concerning computer implemented processes since *Parker* v. *Flook* are correct and are leading to case law continuity. 7

IV. Conclusion 10

TABLE OF AUTHORITIES

CASES: PAGE

In re Chatfield, 545 F.2d 152 (C.C.P.A. 1976) Cert. Denied 434 U.S. 875 (1977) 3

Dann v. *Johnston*, 425 U.S. 219 (1976) 3, 7

Diamond v. *Chakrabarty*, _____ U.S. _____, 484 PTCJ D1 (1980) 6

In re Freeman, 573 F.2d 1237 (C.C.P.A. 1978) 7

In re Gelnovatch, 595 F.2d 32 (C.C.P.A. 1979) 9

Gottschalk v. *Benson*, 409 U.S. 63 (1972) .....*passim*

Graham v. *Deere*, 383 U.S. 1 (1966) 9

In re Johnson, 589 F.2d 1070 (C.C.P.A. 1978) 7

In re Maucorps, 609 F.2d 481 (C.C.P.A. 1979) 9

Parker v. *Flook*, 437 U.S. 584 (1978) .....*passim*

In re Sarkar, 588 F.2d 1330 (C.C.P.A. 1979) 9

In re Sherwood, 613 F.2d 809 (C.C.P.A. 1980) 9

United States v. *Adams*, 383 U.S. 39 (1966) 9

In re Walter, _____F2d_____, 205 USPQ 397 (C.C.P.A. 1980) 8

STATUTES:

35 U.S.C. § 101 .....3, 6, 7, 8, 10

35 U.S.C. § 102 10

35 U.S.C. § 103 10

No. 79-855

IN THE

Supreme Court of the United States

OCTOBER TERM, 1979

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS, *Petitioner,*

v.

JOHN J. BRADLEY AND BENJAMIN S. FRANKLIN, *Respondents.*

On Writ Of Certiorari To The United States Court Of Customs And Patent Appeals

---

BRIEF AMICUS CURIAE FOR HALLIBURTON SERVICES

I. THE NATURE AND INTEREST OF THE AMICUS

This brief amicus curiae is submitted by Halliburton Services in accordance with Rule 42(2) of this Court. Letters of consent from both parties have been filed with the Clerk of the Court.

Halliburton Services has as its principal business the provision of field services to the energy industry on a worldwide basis. Halliburton Services provides equip-

ment, technology, personnel and materials to energy related industries; services that would not be economically or practically feasible for these industries to maintain on their own. Field services provided by Halliburton Services include oil well cementing, well stimulation services, formation testing, bulk handling equipment, well logging, environmental control, sea structure services and other special products and services relating to the energy industry.

In the course of its operations in the field service industry Halliburton Services has occasion to develop and use services and instruments which include microprocessors, special purpose digital computers and general purpose digital computer controls and systems. Halliburton Services' development of such systems has given rise to proprietary information and property deemed of great value. Considerable uncertainty has arisen as to the available protection afforded to such property interests under the present patent laws. Halliburton Services appears here in an attempt to assist the Court in clarifying and developing this area of the patent law in an orderly manner.

II. SUMMARY OF THE ARGUMENT

A. The quest by the Patent and Trademark Office for a *per se* rule of unpatentability for any computer implemented process, however claimed, is not correct and not in the interest of justice.

B. The Court of Customs and Patent Appeals decisions since *Parker* v. *Flook*, 437 U.S. 584 (1978) on computer implemented processes are based on decisions of this Court. These decisions are correct, logically consistent and are leading to a continuity of case law in this area.

III. ARGUMENT

A. The Patent And Trademark Office Quest For A *Per Se* Rule Of Unpatentability For Any Computer Implemented Process, However Claimed, Is Not Correct And Not In The Interest Of Justice.

The present case and its companion case of *Diamond* v. *Diehr, et al,* (No. 79-1112) represent the fifth and sixth attempts[1] by the Patent and Trademark Office (hereafter PTO) in the past ten years to obtain from this Court a *per se* rule of unpatentability for computer implemented processes under 35 USC 101. Petitioner has also filed application for Writ of Certiorari on the same basis in the related case of *Diamond* v. *Sherwood,* (79-1941), thus making a total at the present time of seven such attempts to obtain a *per se* rule of unpatentability. The present multiple attack (three simultaneous cases) represents the most extensive effort to date by Petitioner and is unprecedented.

This Court has very wisely declined to fashion such a *per se* rule of unpatentability[2] based on the facts of *Benson, Johnston, Chatfield,* or *Flook.* Your amicus respectfully submits that once again, based on the facts

---

[1] Previous attempts were made (see petitioner's briefs) in *Gottschalt* v. *Benson,* 409 U.S. 63 (1972); *Dann* v. *Johnston,* 425 U.S. 219 (1976); *In re Chatfield,* 545 F2d 152 (C.C.P.A. 1976), Cert. denied 434 U.S. 875 (1977); and *Parker* v. *Flook,* 437 U.S. 584 (1978).

[2] That this Court specifically declined to fashion such a *per se* rule was eloquently stated in *Benson,* 409 U.S. 63,71 by Mr. Justice Douglas:

> "We do not hold that no process patent could ever qualify if it did not meet the requirements of our prior precedents. It is said that the decision precludes a patent for any program servicing a computer. We do not so hold."

of the present case, this Court should decline to fashion a *per se* rule of unpatentability for any process, however claimed, which may be implemented by or involving a digital computer.

In *Benson*[3], and as reiterated in *Flook*[4], this Court carefully defined its use of the term "algorithm" to mean a mathematical formula or principle as distinct from the more general term "process". The PTO has consistently, since *Benson*, attempted to expand the term "algorithm" to encompass any process involving a digital computer. Petitioner now labels (Pet. Br. 14) the C.C.P.A.'s careful distinction of the term "algorithm" in the mathematical sense (as distinct from a non-mathematical computer implemented process) in the instant case, as "basically illusory" because a digital computer is said to be designed "solely to perform mathematical computations". This line of thought reflects a gross oversimplification which is clearly in error.

Digital computers are designed to execute a defined instruction set, only a few of which usually involve the basic mathematical operations of addition, subtraction, multiplication and division. Most of the instruction set of a digital computer are conditional logic instructions, control instructions and instructions for moving data around in its memory. To characterize such operations

---

[3] *Gottschalk* v. *Benson*, 409 U.S. 63,65

"A procedure for solving a given type of mathematical problem is known as an "algorithm"."

[4] *Parker* v. *Flook*, 437 U.S. 584

"We use the word "algorithm" in this case as we did in *Gottschalk* v. *Benson*, 409 U.S. 63,65 to mean (a) procedure for solving a given type of mathematical problem". (footnote)

as "mathematical computations" is as erroneous as characterizing an elephant as a winged mammal.

Such erroneous oversimplifications on the part of the PTO have led to vast confusion and uncertainty on the part of inventors as to the extent of protection offered their inventions under the present patent laws. Practitioners of the state-of-the-art electronic design and development, such as your amicus, have found it to be convenient and economical to use newly available "microprocessor" circuits in designing new equipment which type of equipment was previously designed employing discrete circuit elements (such as transistors, resistors, capacitors, etc.). Under the PTO logic applied to the instant case, a piece of equipment which would clearly have comprised statutory subject matter under 35 USC 101 when designed with discrete circuit elements (for example, a well logging instrument) would be nonstatutory subject matter if redesigned *and improved* using microprocessor chips. This is clearly an absurd result.

State-of-the-art design practictioners presently use discrete circuit elements, microprocessors, and programmed general purpose digital computers virtually interchangeably in new systems under development. Cost, space availability, lead time and convenience will usually dictate which one, or all, of these three equivalent modes are used to perform a given design function or specification. Why then should one design mode comprise statutory subject matter and another not, when they are functionally equivalent?

Petitioner asserts that microprocessors or digital computers programmed to perform specific instructional sequences are not new machines or new uses of existing

machines[5] in the sense of the patent laws because such instructional sequences are "inherently available operation(s) of the already patented machine" (Pet. Br. 18). This absurd proposition assumes that the inventor of the computer or microprocessor correctly foresaw each and every use to which his invention might be employed. Such an assertion is akin to a statement that the compiler of a dictionary is the true author of all the literature in a language, since such literature merely comprises words which he has defined.

Petitioner has pointed out that this Court has twice before[6] expressed that a final resolution of the patentability of computer programs *per se* would better be left to the Congress (Pet. Br. 22). Petitioner fails to recognize however, that this does not answer the question of the patentability under 35 USC 101 of *particular* computer or microprocessor implemented processes or systems being challenged under the present law, such as that of the case at bar. Moreover, the expression of such sentiments by this Court should not be construed as a mandate to enforce a *per se* rule of unpatentability in the intervening period before the Congress acts, for to do so would *de facto* usurp the very Congressional powers being commented upon by this Court in the cited passages from *Benson* and *Flook*.[7] The real problem confronting the courts and the patent bar is to establish viable guidelines in the form of rules of law and tests to

---

[5] Both of which clearly comprise statutory subject matter under 35 USC 101 which states:

> § 101 Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvements thereof, may obtain a patent therefor subject to the conditions and requirements of this title.

[6] *Gottschalk* v. *Benson*, 409 U.S. at 72-73 and *Parker* v. *Flook*, 437 U.S. at 595.

[7] The Court recognized this in its recent decision in *Diamond* v. *Chakrabarty*, _____ U.S. _____, 484 PTCJ D1 at D4 (1980), in stating: "*Flook* did not announce a new principle that inventions in

determine the patentability of particular computer implemented processes under the present patent laws[8]. Your amicus respectfully submits that the Court of Customs and Patent Appeals decisions in this area since *Parker* v. *Flook*, *supra*, have made an excellent start in this process.

B. The Court Of Customs And Patent Appeals Decisions Concerning Computer Implemented Processes Since *Parker* v. *Flook* Are Correct And Are Leading To Case Law Continuity

Since this Court's decision in *Parker* v. *Flook*, *supra*, the C.C.P.A. has developed a logical two step test[9] for determining patentability of statutory subject matter under 35 USC 101. This test was developed based on the decisions of this Court in *Benson*, *Johnston* and *Flook*. The test may be stated as follows[10]:

1. Determine whether the claim directly or indirectly recites steps which are themselves calculations, formulae or equations (i.e. algorithms); and, if so

2. Further analyze the claim taken as a whole to ascertain whether the claim merely recites a

---

areas not contemplated by Congress when the patent laws were enacted are unpatentable *per se*."

[8] Apparently petitioner reluctantly concedes that some computer implemented processes conceivably could be patentable (See Petitioner's Brief footnote 22, p. 15) even while seeking what amounts to a *per se* rule against their patentability.

[9] The two step test was first stated by the C.C.P.A. in *In re Freeman*, 573 F.2d 1237 (C.C.P.A. 1978) and first applied in its entirety in *In re Johnson*, 589 F.2d 1070 (C.C.P.A. 1978) after this Court's decision in *Flook*.

[10] *In re Johnson*, *supra*.

mathematical formula or a method of calculation as in *Benson* and *Flook*.

The second step of this test has been further refined by the C.C.P.A.[11] to include a determination of whether any algorithm found in the first step is implemented in a specific manner to define structural relationships between the physical elements of the claim (in apparatus claims) or to refine or limit claim steps (in process claims), the claim being otherwise statutory, it meets the requirements of 35 USC 101 for statutory subject matter.

A key element of the C.C.P.A. two step claim analysis test is that the *claim as a whole* is analyzed[12]. This principle is explicitly stated as being correct by this Court[13] in *Flook*. In the instant case, however, Petitioner urges a "point of novelty" test for dissecting a claim and discarding a portion of the claim (Pet. Br. 9), which is purported to be based on *Flook*. A careful reading of the *Flook* claim analysis as set out by this Court however, compels your amicus to conclude that the Petitioner is simply in error in this interpretation[14].

Petitioner's contention that the claim analysis outlined by this Court in *Flook* stands for the proposition that the respondent's "idea" (Pet. Br. 12), in ef-

---

[11] *In re Walter*, _____ F2d _____, 205 U.S.P.Q. 397 (C.C.P.A. 1980)

[12] *In re Walter, supra, In re Johnson, supra.*

[13] *Parker* v. *Flook*, 437 U.S. at 594 and n. 16.

[14] *Parker* v. *Flook*, 437 U.S. at 594 states:

> "Respondents process is unpatentable under § 101 not because it contains a mathematical algorithm, but because once that algorithm is assumed to be in the prior art, the application, *considered as a whole*, contains no patentable invention." (Emphasis supplied)

fect, be used as prior art against him when implemented by a computer and be disregarded in analyzing the remainder of the claim, results in a hindsight approach to the issue of patentability. This Court has many times condemned such hindsight approaches, most recently for example, in the cases of *Graham* v. *Deere*, 383 U.S. 1 (1966) and *United States* v. *Adams*, 383 U.S. 39 (1966). This contention also completely ignores the fact that this Court's *Flook* claim analysis was applied to mathematical algorithms or principles of nature and not to any inventive concept, in general, which could be implemented on a computer or microprocessor. In short, your amicus is forced to conclude that Petitioner's "point of novelty" test results in a serious distortion of this Court's and other courts' prior holdings, and should thus be discarded.

Since its adoption of the two step test as set out in *In re Freeman, supra*, the C.C.P.A. has applied this test in a consistent and even handed manner. Four C.C.P.A. decisions[15], since *Flook*, have upheld the PTO Board of Appeals, while four[16] decisions have reversed the PTO Board of Appeals. These decisions have been very valuable to practitioners, such as your amicus, of state-of-the-art systems design in clearing away the fog of uncertainty as to what type of computer implemented inventions would be protectable under the present patent laws. Perhaps not every question has been, or can be, resolved solely by the application of this test. Perhaps this court will see fit in the instant case, to clarify or

---

[15] *In re Maucorps*, 609 F.2d 481 (C.C.P.A. 1979); *In re Gelnovatch*, 595 F.2d 32 (C.C.P.A. 1979); and *In re Sarkar*, 588 F.2d 1330 (C.C.P.A. 1979); *In re Walter, supra.*

[16] *In re Johnson, supra*; the instant case and its companion case at bar, *Diamond* v. *Diehr, et al*; and *In re Sherwood*, 613 F.2d 809, (C.C.P.A. 1980) Pet. for Cert. pending.

modify this test further. However, the present C.C.P.A. test and the cases applying it does begin to illuminate a path through the darkness of the uncertainty surrounding this complex issue. This area of the patent law should be allowed to develop, supported by the decisions of this Court and the C.C.P.A., on a case by case basis in an orderly manner. The adoption of a *per se* rule of unpatentability for any computer implemented process or the adoption of Petitioner's proposed "point of novelty" test might be administratively expedient to the PTO, but would result in injustice and the furtherance of vast confusion.

IV. CONCLUSION

Your amicus expresses no opinion on the novelty or unobviousness in the sense of 35 USC 102 and 103 of the subject matter of the instant case. The C.C.P.A. has, however, correctly decided that patentable subject matter, in the sense of 35 USC 101, is claimed in the case at bar. This Court should recognize the correctness of this decision, and in so doing, that a class of inventions exists under the present patent laws which may be implemented by means of programmed digital computers or microprocessors.

Respectfully submitted,

WILLIAM JAMES BEARD

Attorney for Amicus Curiae, Halliburton Services

Of Counsel:

JOHN H. TREGONING

Supreme Court, U. S.
FILED
JUL 31 1980
MICHAEL RODAK, JR., CLERK

Nos. 79-855, 79-1112
IN THE

Supreme Court of the United States

October Term 1979.

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS,

Petitioner,

vs.

JOHN J. BRADLEY and BENJAMIN S. FRANKLIN.

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS,

Petitioner,

vs.

JAMES R. DIEHR, II, and THEODORE A. LUTTON.

BRIEF AMICUS CURIAE OF LOS ANGELES PATENT LAW ASSOCIATION.

REED C. LAWLOR,
2500 East Colorado Blvd., Suite 350,
Pasadena, Calif. 91107,
Attorney for Amicus Curiae.

Of Counsel:
JAMES W. GERIAK,
LYON & LYON,
Ninth Floor,
800 Wilshire Blvd.,
Los Angeles, Calif. 90017.

Parker & Son, Inc., Law Printers, Los Angeles. Phone 724-6622

TABLE OF CONTENTS

Page

I.

The Question 1

II.

Consent 2

III.

Interest of Amicus 2

IV.

Statutes Involved 3

V.

Short Statement of Argument 3

VI.

Petitioner Has Mischaracterized the Issues in the Two Subject Cases, Since Neither Respondent Has Claimed a Program Per Se. The Real Issue Is Presented Above by Amicus 5

VII.

There Is Ample Judicial Precedent for Upholding the Decisions of the Court Below and No Judicial Precedent That Requires Reversing Those Decisions .. 7

VIII.

Title 35 of the United States Code Authorizes Patents for Program-Implemented Inventions 13

IX.

The Claims of the Respondents, Bradley and Diehr, Satisfy the Requirements of 35 U.S.C. 112, and Therefore, Define Physical Things Not Non-Physical Things 15

X.

Ideas and Mathematics Are Not Protected by Patents, but Only Inventions. Likewise Ideas Are Not Protected by Copyrights, but Only Certain Expressions 18

Page

XI.

The Art of Computer Programming Is of Ancient Origin 20

XII.

Programming a Computer Cannot Convert a Statutory Computer Into a Nonstatutory Computer 24

XIII.

Other Arguments of Petitioner Are Also Specious 25

XIV.

The Time Is Ripe for This Court to Declare That Program-Implemented Inventions Constitute Statutory Subject Matter 28

XV.

Conclusion 30

TABLE OF AUTHORITIES CITED

U.S. Cases: Page

Bernhart and Fetter, Application of, 417 F.2d 1395 (1969) 16, 25

Chatfield, Application of, 545 F.2d 152 (1976) cert. den., 434 U.S. 875 15

Data Cash System, Inc. v. JS & A Group, Inc., 480 F. Supp. 1063 (N.D. Ill. 1979) 10

Decca Limited v. U.S., 544 F.2d 1070, 188 USPQ 167 (Ct. cl. 1975) 12, 13

Deepsouth Packing Co. v. Laitram Corp., 406 U.S. 518 (1972) 15, 26

Diamond v. Chakrabarty .... U.S. ...., Opinion No. 79-136 (1980) 15, 16

Eibel Process Company v. Minnesota Ontario Paper Company, 261 U.S. 45 (1923) 17, 25

Faulkner v. Gibbs, 388 U.S. 267 (1949) 10, 16, 17

Gibbs v. T.Z.R. Amusement Corporation, 14 F. Supp. 957 (1936) 10, 16

Gottschalk v. Benson, 409 U.S. 63 (1972) 4, 7, 8, 12, 16, 17, 29

Lang, Ex parte, 56 USPQ 423 (1943) 10

MacKay Radio and Television Co. v. Radio Corp., 306 U.S. 86 (1939) 16, 25

Parker v. Flook, 437 U.S. 584 (1978) 4, 7, 8, 15, 16, 17, 18, 29

The Great Atlantic and Pacific Tea Company v. Supermarket Equipment Corp., 340 U.S. 147 (1950) 29

Walter, Application of, 618 F.2d 758 (1980) 24

White-Smith Publishing Co. v. Apollo Co., 209 U.S. 1 (1908) 9, 10

Page

Statutes:

35 USC 101 ... 1, 3, 10, 14, 15, 16, 17, 19, 24, 25, 26
35 USC 102 ..... 2, 3, 15, 18, 25, 29
35 USC 103 ..... 2, 15, 25, 29
35 USC 112 ..... 1, 3, 9, 15, 16, 17, 19, 23
17 USC 102(b) (1976 Act) 19
17 USC 103(d) (1976 Act) 10
17 USC 117 (1976 Act) 19, 25
17 USC 301(d) (1976 Act) 10, 19

Proposed Legislation:

S.1042, Sec. 106 (89th Cong., 2nd Sess.) 13, 14

Foreign Cases:

In the Matter of AN APPLICATION by INTERNATIONAL BUSINESS MACHINES CORPORATION for the Revocation of Letters, Patent No. 1,352,742 in the FREDERICK NYMEYER (decided October 16, 1978 by the Royal Courts of Justice of the United Kingdom and not appealed), ... RPC 28

Siemens AG v. Allegemeine Electrizitats-Gesellschaft AEG-Telefunken, (decided June 7, 1977 by the Supreme Patent Court of West Germany), International Review of Industrial Property and Copyright Law, 363, published in English by the Max Plank Institute for Foreign and International Patent, Copyright and Competition Law (Munich) 28

Rules:

Rules of the Supreme Court of the United States, Rule 36 2

Miscellaneous:

Bishop, Morris, "Pascal, The Life of Genius", (Reynal & Hitchcock, N.Y., 1936), p. 28 22

Page

Cohen, Morris R. and Krabkin, I.W., "A Source Book in Greek Science", (McGraw-Hill, 1948) 20

Deller's Walker on Patents, Second ed. (1964) 19

Drachmann, A.G., "Ktesibios Philon and Heron, A Study in Ancient Pneumatics", (Ejnar Munksgaard-Copenhagen, 1948), pp. 16ff 20

Goldstine, Herman H., "The Computer from Pascal to von Neumann", (Princeton University Press, 1972) 20

Gwilt, Joseph, "The Architecture of Marcus Vitruvius Pollio", (John Weale, London, 1860), pp. 249-251 20

Harvard Staff of the Computer Laboratory, "A Manner of Operation for the Automatic Calculator, Vol. I", (Harvard University Press, 1946) 20

Merzbach, Uta, "Georg Scheutz and the First Printing Calculator", Smithsonian Studies in History and Technology, Number 36, (Smithsonian Institution Press, 1977) 22

Morrison, Philip and Emily, "Charles Babbage and His Calculating Engines", (Dover, 1961) 22

Needham, Joseph, "Science and Civilisation in China", Vol. 4, Part 2, (Cambridge University Press, 1965), pp. 281-286 20

Nimmer on Copyright, Sec. 2.03[b] 19

Phylarchus, "Ancient Monopolies", XIV JPOS 348 .. 26

Prager, Frank D., "Examination of Inventions from the Middle Ages to 1836", XLVI JPOS 268 (1964) 22

Page

Price, Derek de Solla, "An Ancient Greek Computer," pp. 60-67, Scientific American, June 1959; "The Tower of the Winds", pp. 387-396, National Geographic, Vol. 131, April 1967; "Gears from the Greeks", Trans. Am. Phil. Soc., Vol. 64, November 1974 20

Report of the President's Commission on the Patent System, November 17, 1966 ..... 13, 14

Rohde, Alfred, "Die Geschichte der Wissenschaftlichen Instrumente vom Geginn der Renassance bis zum Ausgang des 18. Jahrhunderts, (Leipzig, Klinkhardt & Biermann, 1923) 21

Rosenberg, Jerry M., "The Computer Prophets" (MacMillan, 1969) 27

Schuyler, William E. Jr., "Protecting Property in Computer Software", reprinted from The Law of Software (George Washington University, 1969) in 3 Computer Law Service, Sec. 4-1, Art. 4 14

Singer, Charles, et al., "A History of Technology", Vol. III, (Oxford at the Clarendon Press, 1957), pp. 601-603 20

Stibitz Patent No. 2,668,661 23

Eckert Mauchly Patent No. 3,120,606 23

Nos. 79-855, 79-1112
IN THE

Supreme Court of the United States

October Term 1979.

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS,

Petitioner,

vs.

JOHN J. BRADLEY and BENJAMIN S. FRANKLIN.

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS,

Petitioner,

vs.

JAMES R. DIEHR, II, and THEODORE A. LUTTON.

BRIEF AMICUS CURIAE OF LOS ANGELES PATENT LAW ASSOCIATION.

I.

The Question.

Whether the subject matter of a patent claim that specifies, in accordance with the requirements of 35 USC 112, means or steps for performing specified functions is non-statutory under 35 USC 101 merely because one or more of the means or steps may be implemented by the programming of a digital processing device.

II.

Consent.

This Brief Amicus Curiae is submitted on behalf of the Los Angeles Patent Law Association in support of Appellant Bradley, *et al.* and Appellant Diehr, *et al.* under Rule 36 of this Court. The consents of counsel of the various parties have been obtained to the filing of this brief and have been separately filed with the Court.

III.

Interest of Amicus.

The Los Angeles Patent Law Association is an organization of lawyers and patent agents primarily representing inventors in Southern California, where a large amount of computer development occurs along with a large amount of development of programs and their applications. This association appears here in an attempt to assist the Court and in an effort to assure orderly development of the patent law in the modern technological era in which programming of computers plays an important role. More particularly, Amicus is deeply concerned that Petitioner's briefs as a whole badly confuse the issues.

Much of the confusion arises because the language used by Petitioner has different meanings under different circumstances, and care has not been exercised by Petitioner to distinguish these meanings according to those circumstances.

Amicus has no interest in the patent applications of Respondents and does not consider itself prepared to form a judgment as to whether or not the inventions described and claimed there meet the standards of Sections 102 and 103.

It is the position of Amicus that Petitioner has mischaracterized the facts of the cases before the Court and has also

distorted much of the law, out of its zeal to obtain a particular ruling on an issue which is not even present in the cases before the Court.

It is also the position of Amicus that a properly claimed invention is not excluded from the classes of inventions recited in 35 USC 101 merely because it makes use of a program modified computer, and that, therefore the decisions of the court below should be affirmed.

IV.

Statutes Involved.

Section 112, paragraph 6, provides that:

> "An element in a claim for a combination may be expressed as a means or step for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof."

V.

Short Statement of Argument.

Petitioner seems to take the position that no matter what other merit an invention may have, if a claim directed to that invention can be practiced with the aid of a programmed computer that is a program-modified computer then the claim is nonstatutory. Amicus disagrees.

Computers and computer programming are becoming almost as common today as gears and valves. There is no valid reason for denying patentability to inventions that employ computers or programming under the present law than there is for denying patentability to inventions that employ gears or valves.

The main thesis of this brief by amicus is that programmed computers are modified physical things and remain physical

things after such modification and that programming of computers is also a physical thing or process. Such physical things, may be patentable in themselves or form parts of patentable combinations.

Writing a program is not the same thing as programming a computer. Writing a program is a literary effort, though highly technical. But programming a computer involves changing it in a way that makes it perform a new function.

Holdings of *Gottschalk v. Benson*, 409 U.S. 63 (1972), and of *Parker v. Flook*, 437 U.S. 584 (1978), have no bearing on these cases because there was no novel step in either of them that could not be carried out mentally, that is without physical aids. Furthermore, neither of them explicitly claimed a program-modified computer.

The Petitioner does not face the issue presented by the claims but attempts to obscure the issue by using ambiguous terms without clarification. He does not even distinguish a program *per se* written on a piece of paper which is merely literary property and a programmed computer which is a physical thing that embodies an invention.

Petitioner urges this Court to focus its attention on computer implemented *programs*. But the focus should be on program-modified *computers*. Petitioner is looking through the wrong end of the telescope.

Programmed computers are members of an old, even ancient, art that lies within the scope of the present patent laws of the United States.

The present patent law sanctions patents on inventions that involve programmed computers or modifying computers with programs.

VI.

Petitioner Has Mischaracterized the Issues in the Two Subject Cases, Since Neither Respondent Has Claimed a Program Per Se. The Real Issue Is Presented Above by Amicus.

It is important to distinguish several different concepts. First, an idea; second, an expression of the idea; third, a program *per se*; fourth, a programmed machine; fifth a process for operating a machine; sixth, a machine or process that utilizes a programmed computer; and seventh, a combination that may utilize a program-modified computer as well as other physical things. The most unruly and ambiguous terms used frequently by the Petitioner in his brief are "expression" and "program". But worst of all, nowhere does Petitioner recognize that programming of a computer modifies it thereby causing it to function in a different way. And this is the most essential fact of all.

Patentability of programs *per se* is not at issue in these cases. But the question defined by Amicus is at issue. The gap between the claimed inventions of Respondents and their ideas is a wide one.

Petitioner has presented the issues in *Bradley, et al.* and *Diehr, et al.* and also in the upcoming case *Diamond v. Sherwood*, No. 79-1941, with substantially the same stereotyped question, namely, "Whether a computer program . . . is patentable subject matter under 35 USC 101." In each case the blank part of the issue-defining clause makes reference to the general subject of the Respondent's invention. There are thousands of such general subjects. But in all these cases, Petitioner asks whether a "program" is patentable even though none of the Respondents claims a program, but only claims means or steps that may be implemented with a program-modified computer.

The issues framed by Petitioner are almost meaningless in these cases because the term "computer program" is subject to wide interpretation, none of which, however, justifies Petitioner's assertion that the Respondents are claiming a computer program. The failure of Petitioner to exercise care in the use of such terms as "computer program" and "expression" obscures the meaning and purpose of his arguments.

The inventions for which patents are sought by Respondents are defined by the claims, not by the conclusions of the Examiner and the Board, which only mischaracterize the inventions; and by the overly-broad questions upon which Petitioner appears to seek rulings by this Court.

Any broad statement that computer programs are or are not patentable could lead to grave misunderstanding and such a statement would not decide whether the claims presented in these cases are patentable. Hence Petitioner's definitions of the issues are inapposite.

Considering the loose manner in which the Petitioner employs various terms, he could interpret a negative answer to the questions presented by him as a signal to reject all claims to apparatus or processes that could be practiced in part or in whole by modifying a computer in any way whatsoever. This follows since Petitioner's framing of the question as he has, is tantamount to a representation that an answer would be dispositive of the *Bradley, et al.* and *Diehr, et al.* cases. But this is not really the case. The Court is implored to focus its attention on the claims before it rather than on the overly-broad question framed by Petitioner and the loosely worded arguments of Petitioner.

VII.

There Is Ample Judicial Precedent for Upholding the Decisions of the Court Below and No Judicial Precedent That Requires Reversing Those Decisions.

There is a common factor in *Gottschalk v. Benson, supra* and in *Parker v. Flook, supra* that has hardly been mentioned explicitly. That factor involves mental processes.

In *Benson*, an attempt was made to claim a mathematical algorithm per se completely apart from any apparatus. As a result, those claims read on a mental process. In holding Benson's claims non-statutory, this Court pointed out that the *claimed* invention "could be performed . . . without any apparatus" at all. *Gottschalk v. Benson, supra*, 68. In other words, this Court realized that the process claimed by Benson could be infringed by mental steps alone and hence, the claim would "preempt the mathematical formula or algorithm itself." The reference to "reentrant register" in Benson's claim 8 does not save the claim from the criticism just mentioned. As correctly pointed out in footnote 26 on page 25 of the Benson Petitioner's Brief

> "Claim 8 makes reference to the use of shift register at one stage of the claimed procedure. The Patent Office Board of Appeals made an administrative determination that this language did not restrict the use of the procedure to digital computers, because the term 'shift register' comprehended a piece of paper with pencilled notations on it."

Amicus has examined the Benson record and agrees that in this determination by the Benson petitioner was correct.

And in *Flook*, the critical element of the claim also was broad enough to read on mental operations. That element read,

> "Determining a new alarm base B1 using the following *equation* . . ." (Emphasis added).

There was nothing in *Flook* to limit the "determining step to a physical process.

Petitioner inpliedly admits that Benson and Flook do not constitute judicial authority requiring the denial of the claims of either Respondent. Instead Petitioner merely asserts (Pet. Br. p. 11) only on the "suggestion" that he attributes to this Court in the *Benson* and *Flook* cases that traditional computer programs . . . *i.e.* software . . . are unpatentable.

But what is a program? Petitioner explains (Pet. Br.* p. 8) that "computer programs . . . are simply the *expression* in language intelligible to the computer of the fundamental ideas of the computer user." (Emphasis added). By this definition a program is something external to a computer. Petitioner's definition becomes an absurdity when one attempts to apply it to the claims of the cases at hand. Anyway, it is confusing since computers are not "intelligible" except in a metaphoric sense. And at that, they can't understand a "user's ideas" as alleged by petitioner.

In any event, nowhere in the claims of *Bradley, et al.* or *Diehr, et al.* does one find any reference to an "expression in language intelligible to the comupter" of anything.

It is true that a computer program may be represented by a flow chart or a block diagram which, indeed, is an expression which is *understandable to a programmer* as to how to program a computer (see, for example, Bradley, J.A. 114-120 and Diehr, App. to Pet. for Cert. pp. 55a-56a.). A computer program may also be expressed in the form of a listing (see Diehr Pet. Br. pp. 8 and 19) or as a set of states of memory cells of a computer memory.

---

*Where an abbreviation Pet. Br. or the like is employed, it refers to the Petitioner's Brief in the *Bradley, et al.* case, unless otherwise indicated.

Flow charts and program listings are "expressions" and both are copyrightable, but neither is patentable.

Under 35 USC 112 elements of claims are also "expressions" and certainly, the specification of a patent application is also an expression.

It would be a distortion of the patent law to say that a physical embodiment of an invention is an expression of the underlying idea especially if one were to argue, as the Petitioner does, that a claim that covers an embodiment of the invention is a patent on an expression that gives a monopoly on the idea as alleged!

As is well known, programs based upon such a flow chart or expressed in such a listing may be embodied in a set of punched cards, or in a microprocessor, or in a memory unit of a computer. When embodied in these machine forms, changes of state are involved, making the set of cards the microprocessor, or the memory unit a new "mechanical" device compared to what it was before. Obviously it is new if it functions in a new way.

In *White-Smith Publishing Co. v. Apollo Co.*, 209 U.S. 1, 18 this Court recognized that a perforated piano roll is a part of a machine, in other words a mechanical device. In more modern language it is also machine-readable. So also is a set of cards and a paper tape that has been punched or perforated in accordance with a program. And so also is a microprocessor or a memory unit of a computer in which the states of elements and the effective connections between then have been modified in accordance with a program.

A perforated piano roll contains a program that causes the sets of keys of a player piano to operate sequentially and otherwise to produce a musical effect. But it does not alter the structure of the piano. On the other hand, a deck

of punched cards or the like, when fed to a computer, changes it structurally so that it will function in a new way.

There is little doubt that a program listing corresponding to a set from punched cards, a perforated tape, a microprocessor, or a memory unit of a computer, is copyrightable. And producing a program listing corresponding from such cards, *et al.* would infringe a copyright on the original listing.[1]

Under the authority of *White-Smith Publishing Co. v. Apollo Co., supra* if follows that any physical thing or medium that has been programmed to function in or with a computer is a mechanical device. Any such mechanical device may be patentable under Section 101 or form part of a combination that is patentable under this same section, if properly claimed. Congress has already made it clear that if a computer program is copyrightable that fact does not preclude an inventor from obtaining a patent on an invention which utilizes a corresponding programmed device, that is a mechanical part based upon such program (17 USC 301(d)).

Even punched cards have been patented. The Board of Appeals itself of the Patent Office has been responsible for the issuance of a patent to IBM on cards punched with an extra row of holes and nonholes arranged to control the operation of a computer. *Ex parte Lang*, 56 USPQ 423 (1934).

In *Faulkner v. Gibbs*, 388 U.S. 267 (S.Ct. 1949) (see also *Gibbs v. T.Z.R. Amusement Corporation*, 14 F.Supp. 957, 961 for claim 9 quoted with changes below) this Court upheld claims which were held valid by the court below and

---

[1]*Data Cash Sys., Inc. v. JS & A Group, Inc.*, 480 F.Supp. 1063 (N.D. Ill. 1979) is inapposite since the program had been published without notice before January 1, 1978, the effective date of the new copyright act.

which employed a special hard-wired program to determine the winner of a game. The claim even involved AND gates and OR gates. Such gates operate on mathematical principles. Written in modern language solely by the addition of the terms OR and AND, claim 9 read as follows:

> "9. A game apparatus comprising a a plurality of electrically connected units, each including a game board with a plurality of apertures therein, an an annunicator with a plurality of indicators thereon, electrical contacts adjacent each of said apertures connected in the circuits of said indicators, said indicators and said apertures corresponding in number and subdivided in corresponding groups, whereby when objects are deposited in said aspertures by the players at the several units, corresponding indicators will be energized, a supplementary signal circuit on each of said units, and AND means for holding said signal circuit open until all of the indicators of any group on each of said units have been energized, and for closing said signal circuit when all of the indicators of any unit have been energized, and OR means controlled by the closing of the signal circuit of the winning unit for discontinuing the signals and opening the circuits of the indicators on all other units." (Captalized words AND and OR have been added for emphasis and clarity.)

This claim covered a hard-wired calculator in the form of an electrical system programmed to determine the winner of a race between a number of players of marble games used to play a game that resembles "BINGO." The AND means and OR means functioned according to principles of Boolean algebra just as does a modern day computer. Such algebra could have been used explicitly to describe the invention. The AND and OR means were not called by these names at the time, and no reference was made to algorithms or programs, since such descriptive language had not yet come into common use. But certainly, changing the language to modernize the claims would not render them nonstatutory.

And the Court of Claims has held, after considering the *Benson* case, that claims based upon an analog disclosure was infringed by a combination in which some of the functions specified in the claims were carried out with the aid of a general purpose digital computer, *Decca v. U.S.*, 544 F.2d 1070 (Ct. Cl. 1975).

At page 1080 of the *Decca* case, in holding the patent valid and infringed by the United States Government, the Court of claims said it this way,

> "On these fact, infringement is not avoided by showing that digital instead of analog techniques are used in the accused system . . .
> Nor are defendant's arguments regarding the software any more persuasive. While Gottschalk v. Benson, 409 U.S. 63, (1972) held that computer programs were not patentable per se, claims 1 is clearly directed to a combination of elements and not just to a program. Nor does claim 1, as applied to Omega, read only on the program in the computer. Indeed, as defendant's expert testified at trial, the program accomplishes nothing by itself; it must be mechanized to achieve the navigational functions for which the Omega system is designed. Thus, defendant's argument that the burst filter, the tracking filter and the Kalman filter by which the Omega computer processes the data are 'mere abstractions' having no 'physical' existence is entirely beside the point. The theory on which, for example, the Kalman filter, is based, and the instructions or program for the computer itself, may be nothing nore than a set of equations, but the means for receiving the signals and processing them and the means for performing those instructions and achieving the end result quite clearly have a tangible, physical existence which meets the terms of the claim."

Since the Decca claims were held valid and infringed by the Decca court, clearly then such a claim defines statutory subject matter. The law cannot be otherwise. And now another agency of the United States Government is asking this Court to hold the use of computer programs renders

combination claims non-statutory. It is curious that Petitioner has not seen fit to call this Court's attention to the *Decca* case.

VIII.

Title 35 of the United States Code Authorizes Patents for Program-Implemented Inventions.

That the present patent law sanctions patents for program-implemented inventions is evidenced by the fact that one former Commissioner of Patents (Edward Brenner) failed in his attempt to have all claims that could be infringed by means of programmed computers removed from the field of statutory subject matter by Congress. Mr. Brenner was a member of the President's Commission for reform of the Patent System, while he was Commissioner of Patents.

Upon issuance of the Commission's Report, a so-called patent modernization bill—S.1042 (89th Congress, 2nd Session), sometimes called the President's Bill, was submitted to Congress. This Bill included the following anti-program protection provision:

> "§ 106. COMPUTER PROGRAMS NOT PATENTABLE.
>
> "A plan of action or set of operating instructions, in whatever form presented, to cause a controllable data processor computer to perform selected operations shall not be patentable."

This section was introduced into the bill to carry out a recommendation of the President's Commission[2] which read as follows:

> "The classes of patentable subject matter shall continue as at present, *except*:
>
> . . .
>
> "3. A series of instructions which control or con-

---

[2]See p. 16 of Report of the President's Commission on Patent System, November 17, 1966 (hereinafter referred to as President's Patent System Report).

> dition the operation of a data processing machine, generally referred to as a 'program', shall not be considered patentable regardless of whether the program is claimed as: (a) an article, (b) a process described in terms of the operations performed by a machine pursuant to a program, or (c) one or more machine configurations established by a program." (Emphasis added.)

In support of the recommendations of the President's Commission, the Report[3] offered only following self-serving explanation to lend justification for its recommendation:

> The Senate Judiciary Committee did not approve S.1042. Though many so-called patent modernization bills have been proposed since that time, none has ever again proposed an anti-program protection provision.

William Schuyler, a later Commissioner, made the following comment:

> "The President's Commission to study the patent system recommended that computer programs be *excised*[4] from the patent system". (Italics added.)

The attempt to excise computer programs from the patent system failed. Inventions based upon programs are *already included* in the patent system and therefore constitute statutory subject matter under Section 101 and are patentable if the other conditions and requirements of Title 35 are satisfied.

The Patent Office, having failed to accomplish its goal in Congress through the President's Commission, has brought a series of cases before this Court to ask this Court to do what Congress declined to do.

---

[3]President's Patent System Report, *supra*, p. 13.

[4]Schuyler, William E., Jr., "Protecting Property in Computer Software", reprinted from The Law of Software (George Washington University, 1969) in 3 Computer Law Service, Sec. 4-1, Art. 4.

This Court has no authority to extend the patent laws to cover subject matter not already covered. *Deepsouth Packing Co. v. Laitram Corp.*, 406 U.S. 518 (1972). Nor does this Court have authority to excise from the patent laws subject matter that is already covered. Only Congress has the power to perform such excision. Petitioner should go back to Congress again if it wants computer programs to be excised from the present patent laws. *Application of Chatfield*, 545 F.2d 152, 156 (1976) cert. den. 434 US 875.

Considering that Congress has already declined to excise programs and program implemented inventions from the patent law and Petitioner's persistence in asking this court to do what Congress has declined to do, it is important that this Court clearly declare the meaning of the present law, at least insofar as it applies to program-implemented inventions.

IX.

The Claims of the Respondents, Bradley and Diehr, Satisfy the Requirements of 35 U.S.C. 112, and Therefore, Define Physical Things Not Non-Physical Things.

The question before this Court "is a narrow one of statutory interpretation requiring [it] to construe 35 USC 101". *Diamond v. Chackrabarty, supra, Parker v. Flook, supra.*

This Court agreed *unanimously* in *Parker v. Flook, supra,* that Sections 102 and 103 have no application in determining the proper construction of Section 101. Accordingly, the dicta of the majority in *Parker v. Flook* respecting the lack of novelty in the elements of the claims and the application (called abstract) presented by Flook, have no bearing on the present cases, especially since the Respondents are not claiming mathematical formulas or mathematical algorithms

in any sense of the words and their claims involve physical elements that exhibit novelty.

Section 112 of the Patent Act specifies how claims may be worded. More particularly, Section 112 says that an element in a claim for a combination—which is what is claimed by both *Bradley, et al.* and *Diehr, et al.*—may be "expressed" as a means for performing a specified function or as a step for performing a specified function, without, in either case, reciting specific structure, material or acts in support thereof. There is no exception in Section 112 with respect to a programmed computer or any part thereof, whether claimed alone or as part of a combination. Furthermore, as in the *Decca* case, *supra*, such a claim is to be construed by the Trial Court to cover the corresponding structure, material or act described in the specification, and equivalents thereof.

Claims properly drawn in accordance with Sec. 112 cannot read on mental operations or ideas because the human mind is not a "digital computer or its equivalent." *Application of Bernhart and Fetter*, 417 F.2d 1395 (1969) at 615 and 616. They cover only the physical embodiment of the invention disclosed and its equivalents.

P. J. Feiderico has pointed out that *Gibbs* provides for patenting anything under the sun that is made by man. *Diamond v. Chakrabarty, supra*, note 6. Anything refers, of course, to physical things not just ideas or other non-physical things. The mathematical algorithms and equations of *Benson* and *Flook* are not physical things. But programmed computers are physical things and hence constitute statutory subject matter under Sec. 101. And combinations that involve physical things constitute statutory subject matter even if a property, or relation involved is defined in mathematical terms. *MacKay Radio and Television Co. v. Radio Corp.*, 306 U.S. 86 ((1939); *Faulkner v. Gibbs*,

supra; *Gottschalk v. Benson, supra* (1972); *Parker v. Flook, supra*, or the inventions utilize natural laws that may be described mathematically. *Eibel Process Company v. Minnesota Ontario Paper Company*, 261 U.S. 45 (1923).

In *Benson* and *Flook*, the real difficulty was that the offending steps covered mental acts and thus on their face would render persons who performed such mental acts liable for infringement and there was nothing else new in the physical aspects of the claims.

The claims of *Benson* and *Flook* did not meet the standards of Sec. 112, paragraph 6, but the claims of the present Respondents do.

If the standards of Sec. 112, paragraph 6, are met, then the claims cover physical things, not abstract things such as ideas and mathematical algorithms. The presence of such apparatus and processes in the claims render them statutory under Sec. 101.

Claim 1 which is typical of the *Bradley, et al.* case calls for a multi-programming computer together with first means . . ., second means . . . , third means . . . , and fourth means . . . all for performing specified functions. These claims, therefore, cover only the physical system disclosed by *Bradley, et al.* and equivalents thereof. They do not cover mental steps and cannot be performed without the aid of any apparatus at all as was the case in *Benson*. In the *Diehr, et al.* case, claim 1, which is representative of most of the claims, specifies a method that involves calculating with the aid of a digital computer. The very language of this element eliminates any argument that it covers a mental process. Furthermore, the claim specifies numerous physical steps that are interrelated with frequent temperature measurements and the calculations, all in a way not present in either *Benson* or *Flook*.

Claim 11 is somewhat different. It specifies a determining step which by its terms may be performed mentally. But the physical steps are all interrelated in what appears to be a new way. And even though newness should not be considered except under Sec. 102, this fact distinguishes the claim clearly from any involved in the *Flook* case, where the non-calculating steps were admittedly all old. The determining step of *Diehr, et al.'s* claim 11 really makes the claim complete so that all the other physical steps may be more readily understood, also distinguishing claim 11 from the claims of the *Flook* case.

Amicus believes that all the claims at issue meet statutory standards and define statutory subject matter.

X.

Ideas and Mathematics Are Not Protected by Patents, but Only Inventions. Likewise Ideas Are Not Protected by Copyrights, but Only Certain Expressions.

The Petitioner employs confusing arguments in its attack on the patentability of program-implemented inventions. Petitioner discusses programs as "expressions" of ideas.

While the line between a patentable invention and an idea does not always shimmer with clarity, *Parker v. Flook supra* 456 in the cases at hand, a wide gap, amounting to a legal barrier, lies between a programmed computer or a process of using one and an idea on which it is based. Petitioner does not seem to recognize that this gap exists, but seems to attempt to include a programmed computer under the heading "expression", thus smothering the fact that an invention is far more than an "expression." That gap includes flow charts, program listings, and descriptions.

There is a wide gap between an idea for a program and the implementation of a program in a computer and still a

wider gap between an idea for employing a programmed computer in a combination and the combination itself. There are many lines in that wide gap that separate patentable program-implemented inventions from ideas. The inventions claimed by Respondents lie outside that gap on the patentable implementation side thereof.

The meaning of the term "idea" itself is obscure. Deller defines an idea as an intellectual notion that a thing may be done and if done might be of practical utility. Deller's *Walker on Patents*, Section 101. Respondents' claims would not protect such an idea "but only the structure, material, or acts described in the specification and equivalents thereof.". 35 USC 112, paragraph 6.

The recommendation of Petitioner that program protection be left to the copyright law (Pet. Br. p. 24) because Congress now has such questions under consideration is misleading. What Congress has under consideration is a recommendation that common law rights and state statutory rights of a kind that may now benefit program proprietors fairly explicitly under 17 USC 117, be abolished by repeal of that section in its present form.

Expressions of ideas are copyrightable provided they are works of authorship fixed in a tangible medium of expression. But copyrights on expressions do not give the copyright proprietor a monopoly on his idea. 17 USC 102(b). See also Nimmer on Copyright, Section 2.03[b].

The fact that a written expression of a program may be copyrightable does not preclude allowance of claims that include an element in the form of means or steps for carrying out the function involved. 35 USC 112, 17 USC 102(b) and 301(d). And copyright protection of a program can never substitute for the patent protection of a program-implemented invention.

XI.

The Art of Computer Programming Is of Ancient Origin.

There is a myth circulating throughout the land that algorithms and computer programs involve new concepts which came into existence only after the present patent code was enacted in 1952 and hence lie outside the patent laws. This myth is false.

Beginning about 1939 IBM and later the United States Government supported the development and use of the MARK I computer at Harvard University[5]. In 1943 the United States Government commenced the support of the ENIAC at the Moore School of Engineering at the University of Pennsylvania. Both computers involved programming with electrical devices. But that was not the beginning. Many ancient devices were programmed mechanically to perform needed calculations.

Ancient inventions that involved algorithms, radices, calculations and "programs" included the planetarium of Andronika,[6] the Antikythera device,[7] odometers of ancient China,[8] Greece[9] and Rome,[10] and clypsedrae.[11] The idea

---

[5]Goldstine, *The Computer from Pascal to von Newmann*, Princeton University Press, (1972) pp. 151 ff;

Harvard staff of the Computer Laboratory, "A Manner of Operation for the Automatic Calculator, Vol. I", (Harvard University Press, 1946).

[6]Price, Derek de Solla, "The Tower of the Winds", pp. 387-396 National Geographic, Vol. 131, April 1967.

[7]Price, Derek de Solla, "An Ancient Greek Computer", pp. 60-67 Scientific American, June 1959; and "Gears from the Greeks", Trans. Am. Phil. Soc. Vol. 64, November 1974.

[8]Needham, Joseph, "Science and Civilisation in China", Vol. 4, Part 2, (Cambridge University Press, 1965), pp. 281-286.

[9]Cohen, Morris R. and Drabkin, I. E., "A Source Book in Greek Science, p. 342 (McGraw-Hill, 1948).

[10]Gwilt, Joseph, "Architecture of Marcus Vitruvius Pollio," (John Weale, London, 1860), pp. 249-251.

[11]*Singer, Charles, et al.*, "A History of Technology", Vol. III, (Oxford at the Clarendon Press, 1957), pp. 601-603;

Drachmann, A. G., "Ktesibios Philon and Heron, A Study in Ancient Pneumatics", (Ejnar Munksgaard—Copenhagen, 1948), pp. 16 ff.

behind clypsedra, which were water clocks, was to perform very advanced calculations to indicate the time of day in an era when the lengths of the hours changed from day to day. This need arose because daylight which extended from sunrise to sunset was divided into 12 hours of equal length and nighttime which extended from sunset to sunrise also consisted of 12 hours of equal length. Some of the clypsedra are believed to have been programmed partly by means of gears and rotating charts to perform the needed calculations automatically. It is said that such geared water clocks were also used as late of the 17th century. Other clypsedrae are believed to have been programmed by manually adjusting the rate of flow of water into a vessel by means of a calibrated valve from day to day according to the season. Such programs they had both in machines and in processes. But we know little about them because the Greeks had not learned the value of a patent system for advancing the useful arts.

In about 1580 A.D., Rudolfus II, emperor of the Holy Roman Empire, granted Rückert a patent on a programmed recording pedometer that was used to survey the emperor's land.[12] This paper-tape recorder was programmed to be operated by a pedometer to mark the distances traveled on a narrow strip of paper as the pedometer was carried along the course being surveyed.

Blaise Pascal developed a calculating machine in 1642 for performing arithmetic calculations in different coinage

---

[12]Rohde, Alfred, "Die Geschichte der Wissenschaftlichen Instrumente vom Beginn der Renassance bis zum Ausgang des 18. Jahrhunderts, (Leipzig, Klinkhardt & Biermann, 1923).

systems, *e.g.* French or English.[13] The programs for adapting the calculator to operate with different coinage systems (which operated on different radices) were embodied in different sets of gears. Pascal's calculator even performed a carry operation automatically. Louis XIV granted Pascal a patent on his calculator in 1649 A.D.[14]

Charles Babbage pioneered the development of general purpose computers beginning in the early part of the nineteenth century. Programs consisting of series of written instructions to be embodied in Babbage's mechanical calculators were published in 1840 and 1872.[15] The programs were actually embodied in punched cards that formed parts of his machine.

The Scheutz family actually reduced some of the Babbage inventions to practice in about 1850.[16] One of their calculating machines, patented in Great Britain in 1854, was transferred to the Dudley Observatory at Albany, New York, in 1856, where it was then employed for stamping plates for printing books of mathematical tables beginning in 1857.[17] When it was delivered to Albany, the machine was accompanied by an explanation of the procedure for converting its operation from one number system to another, *i.e.*, from one radix to another.[18]

---

[13]Bishop, Morris, "Pascal, The Life of Genius, (Reynal & Hitchcock, N.Y., 1936), p. 28.

[14]Prager, Frank D., "Examination of Inventions from the Middle Ages to 1836" XLVI JPOS268, 280 (1964).

[15]Morrison, Philip and Emily, "Charles Babbage and His Calculating Engines", (Dover, 1961).

[16]Merzbach, Uta, "George Scheutz and the First Printing Calculator", Smithsonian Studies in History and Technology, Number 36, (Smithsonian Institution Press, 1977.)

[17]Merzbach, *id.*, p. 26. One of the books is the National History Museum of the Smithsonian Institution.

[18]Merzbach, *id.*, p. 28.

Stibitz U.S. Patent No. 2,668,661 for which application was made April 19, 1941, was granted on a digital calculating machine which, among other things, was programmed to operate with radices 2 and 5 and to perform a carry operation. Claim 2 of that patent reads:

> "In a calculating device, a summing means comprising augend relays for representing a numerical value to which another numerical value is to be added, addend relays for representing the latter said numerical value and sum relays for representing the resulting sum of said values, an electrical circuit system for each set of one augend, one addend and one sum relay including incoming carry 0 and carry 1 leads for alternatively electrically characterizing outgoing carry 0 and carry 1 leads, the outgoing carry leads of one set of said relays being connected to the corresponding incoming relays, and means including said incoming carry leads and the augend and addend relays of a set for controlling the sum relay of said set."

The foregoing claim went into more detail than is now required under 35 USC 112.

The Eckert and Mauchly patent 3,120,606 for which application was made June 22, 1947, for the ENIAC computer includes the following claim which makes specific reference to "program means" for performing certain functions of the computer:

> "96. In a multipurpose computer, a multiplicity of pulse responsive units each having an input and an output, at least one unit including means for performing an arithmetic computation on two pulse signals received at a radio frequency rate at its input, said unit including means operative to develop pulse signals at a radio frequency rate at its output to represent the computed result; a source of input data including means for reproducing the input data in the form of pulse signals at a radio frequency rate, *program means* for routing the pulse signals from said source and from the output of at least one of said units to the inputs of the units in a selected order to cause a sequence to be

selected for a particular problem, and output means responsive to the computed result." (Emphasis added).

These are just samples of patents on programmed computers that were old art in 1952 when the present patent law took effect.

Thus we see that programming is an old, indeed ancient, art in the field of computer technology. So it is not reasonable to urge that computer programming is a new technology which does not constitute patentable subject matter because Congress could not have had it in mind when it enacted the Patent Law of 1952.

Any rule that a new technology remains unpatentable until Congress recognizes it, would defeat the purpose of the Constitution. Even so, computer programming would be outside any such rule.

XII.

Programming a Computer Cannot Convert a Statutory Computer Into a Nonstatutory Computer.

Petitioner has presented many arguments to the effect that programming of computers cannot constitute statutory subject matter. Its arguments are based upon strange suppositions about mathematics, radices, mathematical algorithms, and so on.

If the Petitioner's reasoning were to be accepted and carried to its logical conclusion, even computers themselves would be removed from the scope of Sec. 101. *Application of Walter*, 618 F.2d 758, 769.

Petitioner argues that patenting of program-implemented inventions would create monoplies in unpatentable ideas. If that view of the law is sanctioned by this Court it will be argued next that such reasoning applies to all inventions. Hence, Petitioner's arguments that attack program-implemented inventions must be rejected.

Granting that computers themselves constitute statutory subject matter it necessarily follows that if the computer has been programmed, it still remains statutory subject matter. And claims that specify the computer in such programmed condition or the functioning of a computer in accordance with such condition properly define such subject matter.

XIII.

Other Arguments of Petitioner Are Also Specious.

There seems to be no limit to the number of specious arguments that Petitioner has in its arsenal. But it does not have even one good argument. Some of those specious arguments not discussed above are mentioned briefly below.

Petitioner argues (Pet. Br. p. 11) that a programmed computer is not a new machine. In this, it ignores the fact that a machine is new, if the states or conditions of its elements are new. *Bernhart* and *Fetter, supra* 1400. Greater invention is often required in the setting of states of myriads of memory cells of a computer memory unit, than in the setting of an angle between two antenna arms. (*Mackay Radio and Television Co. v. Radio Corp., supra*), or the setting of the location and pitch of a wire (*Eibel Process Co. v. Minnesota Ontario Paper Co., supra*).

Petitioner argues (Pet. Br. p. 16) that an old computer which has been programmed in a new way does not involve a new use of an old machine. This is like saying that a device constructed of parts of an erector set cannot constitute a new device, since all the parts were there from the beginning. As explained by P. J. Federico at 35 USCA 17, the term "new use" of Section 101 "simply indicates that the procedural steps of a patentable process might be old." And even if every part of a combination is old, it may still be a patentable invention, that satisfies Secs. 102 and 103

as well as Sec. 101. *Deepsouth Packing Co., Inc. v. Laitram Corp.*, 406 U.S. 518 (1972)

Petitioner argues (Pet. Br. pp. 8 and 17) that every program that comes into existence and is employed for modifying a computer, was inherent in that computer. That is like saying that all chemical compounds, including microorganisms, that are brought into existence by man are not patentable because they are inherent in nature or that the combination of parts that form a new oil well tool that increases the recovery of oil are inherent in the operations of a machine shop.

The argument of inherency also fails where the program modified computer performs either by itself or in combination with another physical thing a function which the unprogrammed computer could not perform. A newly programmed computer makes use of a new combination of some of the billions of billions of billions, etc. of inherent functions, just like a new chemical compound is a new combination of old atoms.

Petitioner argues (Pet. Br. p. 20) that a computer program is no more patentable than a recipe in a cookbook. That is true only to the extent that the program is only a written expression like the recipe in a cookbook. But the argument does not apply to a program-modified computer. Interestingly, the oldest patent systems known, namely a patent system that is said to have existed in ancient Greece[19] provided for patenting of recipes. Furthermore, recipes satisfy the broad dictionary definition of algorithm.

The Petitioner, in a very natural way, as is common in the computer industry itself, takes an anthropomorphic view

---

[19]Phylarchus, "Ancient Monopolies", XIV JPOS 348.

of computers. At page 8 of its brief it talks about "instructions" that are only understandable by a computer. And at page 13 of its brief, Petitioner highly exaggerates the capability of a computer by saying that it "comprehends" the translation of an idea into computer language. Such language can only serve to confuse the issue. Sets of electrical signals are received by and stored in modern computers and manipulated by them and (usually) new sets of signals are produced. Programmers and other users of computers understand and comprehend the signals and the input signals may come from other machine parts and the output signals may be fed to other machine parts. But computers don't understand or comprehend anything. They only respond and control.

In about the year 996 A.D., it is said, a successful computer was destroyed because it was believed to have supernatural powers.[20] Exaggerating the capability of computers could lead to their destruction of the incentive to expand their use with new programming.

Petitioner's argument that a computer operates necessarily upon mathematical principles has no relevance to the present questions. Pythagoras philosophized in the 6th century B.C. that all existing things can be described by mathematical principles. That philosophy persists, and the patent laws were enacted by men who must have been familiar with that philosophy of the world. Computers have operated on mathematical principles since time immemorial and many of them have been patented. There is no reason to change the rules today.

At page 23, the Petitioner suggests that the relations of the Federal law of the United States to the laws of certain

---

[20]Rosenberg, Jerry M., "The Computer Prophets", (Macmillan, 1969).

foreign countries would be impaired if the decisions of the Court of Customs and Patent Appeals in these cases are upheld. And how about our relations with countries that do grant patents based on program-implemented inventions? Petitioner must know that patent applications which have been rejected as protecting programs by patent examiners in foreigh countries have been held to be patentable by the courts of those countries.[21] Petitioner again is asking this Court to do the work of that is delegated by the Constitution to Congress.

XIV.

The Time is Ripe for This Court to Declare That Program-Implemented Inventions Constitute Statutory Subject Matter.

The Petitioner has brought a parade of cases before this Court in an effort to establish that programs and program-implemented inventions do not constitute statutory subject matter. And these cases continue to be brought here, in spite of the fact that Petitioner has already made several attempts to have such inventions excised from the patent system, once through Congress and then repeatedly by this Court.

Petitioner states (Pet. Br. p. 13) that it has 3,000 more cases, which may be employed to continue that parade. It can be presumed that most of the 3,000 cases still pending before the Patent and Trademark Office contain claims

---

[21]*Siemens AG v. Allegemeine Electrizitäts-Gesellschaft AEG-Telefunken*, (decided June 7, 1977 by the Supreme Patent Court of West Germany) International Review of Industrial Property and Copyright Law, 363, published by the Max Plank Institute for Foreign and International Patent, Copyright and Competition Law (Munich). See also In the Matter of *AN APPLICATION* by *INTERNATIONAL BUSINESS MACHINES CORPORATION* for the Revocation of Letters, Patent No. 1,352,742 in the *FREDERICK NYMEYER* (decided October 16, 1978 by the Royal Courts of Justice of the United Kingdom and not appealed), ... RPC ....

which are even farther from Benson and Flook than the cases now before this Court. It is therefore suggested that the recognition by this Court at this time that the kinds of claims now before this Court constitute statutory subject matter under present law will encourage Petitioner to proceed to examine those cases under Sections 102 and 103 as it should, and avoid further delay in the issuance of patents on such applications. As to those cases that are still pending in the Patent and Trademark Office that come within the holdings of *Benson* and *Flook*, it can be assumed that most can be disposed of by the Examining Staff and the Board of Appeals and the Court of Customs and Patent Appeals without further burdening this Court.

It is not necessary here to find the exact location of any shimmering line between ideas and mathematical algorithms on the one hand and their patentable applications on the other. In the present instances, there is a wide gap which separates unpatentable ideas from patentable program-implemented inventions. Petitioner has obscured the real issue by disregarding the contents of the gap by simply referring to certain kinds of "expressions," *e.g.*, flow charts and listings, and has implied that Respondents seek to patent such expressions. (Pet. Br. p. 14) Nothing could be farther from the truth. Respondents here do not seek to patent flow diagrams or program listings or even computer programs *per se*. They do seek to patent inventions that are not defined in terms of programs but which in practice are implemented by programming of computers.

Petitioner is probably still flinching from criticisms that it has spawned too many patents in the past. *The Great Atlantic and Pacific Tea Company v. Supermarket Equipment Corp.*, 340 U.S. 147, 158 (1950). The Petitioner needs to be assured that properly claimed inventions constitute statutory subject matter even though they may be partly or

even wholly implemented with the aid of programmed computers and that programming of computers to practice an invention does not convert a claimed invention as a whole into an unpatentable idea. Petitioner needs to be assured that there is nothing in the present statutes that excluded program-modified computers from the patent system.

XV.

Conclusion.

It is submitted that there is ample authority for upholding the decisions of the Court below and no authority for reversing those decisions, and that the claims before the Court constitute statutory subject matter under the present statutes. Amicus therefore urges this Court to affirm the decisions of the Court below.

Respectfully submitted,

REED C. LAWLOR,

Attorney for Amicus Curiae.

Of Counsel:
JAMES W. GERIAK.

Supreme Court, U.S.
FILED
MAY 1 1980
MICHAEL RODAK, JR., CLERK

IN THE

Supreme Court of the United States

No. 79-855

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS,
Petitioner
v.
JOHN J. BRADLEY and BENJAMIN S. FRANKLIN
and

No. 79-1112

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS,
Petitioner
v.
JAMES R. DIEHR, II, AND THEODORE A. LUTTON

BRIEF AMICUS CURIAE OF NATIONAL SEMICONDUCTOR CORPORATION

EDWARD S. IRONS
MARY HELEN SEARS
IRONS AND SEARS
1785 Massachusetts Avenue, N.W.
Washington, D.C. 20036
(202) 466-5200

ROBERT P. BESHAR
One New York Plaza
New York, New York 10004
(212) 558-0206

WILSON - EPES PRINTING CO., INC. - 789-0096 - WASHINGTON, D.C. 20001

TABLE OF CONTENTS

	Page
INTEREST OF THE AMICUS CURIAE	1
STATEMENT OF THE CASE	3
ARGUMENT	5
I. THE *FLOOK* ANALYSIS IS CORRECT	5
A. 35 U.S.C. § 101	5
B. Apposite Decisions	9
1. Cases Concerning "Ideas"	10
2. Combination Claims	12
C. The *Flook* Analysis and § 103	14
D. The *Flook* Analysis and the "Art" of Claim Drafting	15
II. THE CLAIMS AT ISSUE	16
III. THE CCPA DECISIONS	18
A. Disregard of § 101's "New" Imperative	18
B. The *Flook* Analysis and the Subject Matter as a "Whole"	21
C. "Debilitation" of the Patent System	21
CONCLUSION	24

TABLE OF AUTHORITIES

Cases Page

Anderson's-Black Rock, Inc. v. *Pavement Salvage Co.*, 396 U.S. 57 (1969) 11, 12, 15, 17
Application of Bergstrom, 427 F.2d 1394 (CCPA 1970) 6
Application of Diehr, 602 F.2d 982 (CCPA 1979) .. 5, 18, 19, 20, 21
Application of Walter, —— F.2d —— (CCPA 1980) 17, 18, 20, 21, 22, 23
Avery Products Corp. v. *Morgan Adhesives Co.*, 496 F.2d 254 (6 Cir. 1974) 13
Bauer Patent Corp. v. *Westinghouse Electric Corp.*, 193 F.Supp. 868 (D.N.C. 1961) 13
Brenner v. *Manson*, 383 U.S. 519 (1966) 16
Corning v. *Burden*, 56 U.S. (15 How.) 252 (1853) .. 9
Dann v. *Johnston*, 425 U.S. 219 (1976) 15
Eibel Process Co. v. *Minnesota & Ontario Paper Co.*, 261 U.S. 45 (1923) 23
Funk Bros. Seed Co. v. *Kalo Inoculant Co.*, 333 U.S. 127 (1948) 10, 11, 14, 23
Gottschalk v. *Benson*, 409 U.S. 63 (1972) 7, 8, 9, 10
Graham v. *John Deere Co.*, 383 U.S. 1 (1966) 5, 15, 21
In re Bergy, 596 F.2d 952 (CCPA 1979) 19
In re Sherwood, 613 F.2d 809, 204 USPQ 537 (CCPA 1980) 22
Lincoln Engineering Co. v. *Stewart-Warner Corp.*, 303 U.S. 545 (1938) 12, 13, 14, 16, 17
Mackay Radio & Telegraph Co. v. *Radio Corp. of America*, 306 U.S. 86 (1939) 22, 23
Nickola v. *Peterson*, 580 F.2d 898 (6 Cir. 1978) 19, 20
Parker v. *Flook*, 437 U.S. 584 (1978) ..... *Passim*
Pierce v. *Ben-Ko-Matic*, 310 F.2d 475 (9 Cir. 1962) 13
Reiter v. *Sonotone Corp.*, 442 U.S. 330, 60 L.Ed.2d 931 (1979) 3, 18
Rohr Aircraft Corp. v. *Rubber Teck, Inc.*, 266 F.2d 613 (9 Cir. 1959) 13
Sakraida v. *Ag Pro*, 425 U.S. 273 (1976) 15
Weinberger v. *Hynson, Westcott & Dunning*, 412 U.S. 609 (1973) 18

TABLE OF AUTHORITIES—Continued

Statutes	Page
1952 Patent Act	9, 11
R. S. 4886	6
United States Code	
35 U.S.C. § 101	*Passim*
35 U.S.C. § 102	6, 20
35 U.S.C. § 103	12, 14, 15, 18, 20, 21
Miscellaneous	
House Report No. 1923, 82d Cong., 2d Sess. (1952)	6, 15
Manual of Patent Examining Procedure § 706.03 (j)	13
Patent Law Codification and Revision, Hearings Before the Committee on the Judiciary, U.S. House of Representatives on H.R. 3760, 82d Cong., 1st Sess. (June 13-15, 1951)	11
Senate Report No. 1979, 82d Cong., 2d Sess. (1952)	6, 15

IN THE

Supreme Court of the United States

No. 79-855

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS,

Petitioner

v.

JOHN J. BRADLEY and BENJAMIN S. FRANKLIN

and

No. 79-1112

SIDNEY A. DIAMOND, COMMISSIONER OF PATENTS AND TRADEMARKS,

Petitioner

v.

JAMES R. DIEHR, II, AND THEODORE A. LUTTON

BRIEF AMICUS CURIAE OF NATIONAL SEMICONDUCTOR CORPORATION

INTEREST OF THE AMICUS CURIAE

The Court of Customs and Patent Appeals (CCPA) appears determined to eviscerate this Court's holding in *Parker* v. *Flook*, 437 U.S. 584 (1978) in order to begin the process of patenting computer software. Such a campaign is of grave concern to segments of the electronics industry. The electronic storage and manipula-

tion of data is probably the most rapidly expanding industry in the world, with the United States presently the dominant manufacturer.

National Semiconductor Corporation is one of the world's largest manufacturers of integrated circuits.

Historically, there was a separation between the manufacturers of electronic components, such as National Semiconductor Corporation, and the manufacturers of electronic equipment, such as AT&T and IBM. As the components have become more complex and compact, the lines of demarcation between the designers of integrated circuits and the designers of electronic systems have become blurred. The more data which can be stored and manipulated on a single integrated circuit, the smaller, faster, lighter and cheaper become the systems which use the circuits, whether the applications be in computers, missiles, telecommunications or television. Manufacturers of integrated circuits confidently predict that by 1985, they will be able to manufacture for less than $1,000 a single microprocessor integrated circuit with the computing power of a mainframe computer which last year sold for over $1,000,000. Work is well underway on integrated circuits capable of storing a million bits of data on a single silicon chip.

Predictably, the equipment manufacturers have applied enormous pressure through the patent bar on the Patent Office, the courts and the Congress to get patent protection for software. Even though they have prospered mightily without software patent protection, they argue that they will be unwilling or unable to commit funds for future research and development unless granted patent monopolies which assure recapture of those funds. The record does not support this threat. It is the high rate of technological change and the intense competition within the industry which have produced electronic solutions so efficient that users are unable to resist changing

from older non-electronic systems. Companies like National Semiconductor Corporation have literally forced their way into new businesses, such as hand-held calculators, with products so powerful and cheap, so efficient and convenient, that older systems have had to give way. The essence of the latest, advanced, powerful calculators has been the conversion to hardware of what was previously deemed software. To permit the patenting of software would be to stifle that process of technological change and competition because it would radically alter the standard of invention.

Few laymen, lawyers or legislators understand the technical jargon of electronics. The result has been that few understand that software is but a set of instructions used to control a machine. It is true that software is written in languages not readable by any except those trained in the field. It is true software often involves prodigious effort. However opaque the language to laymen, however prodigious the effort to produce, software remains no more an invention than Shephard's Citations, which to non-lawyers are equally opaque and require prodigious efforts to create.

STATEMENT OF THE CASE

The "Question Presented" in each of these cases asks whether the claims at issue define "patentable subject matter under 35 U.S.C. 101". To resolve that question, the Court must determine not only whether the claimed subject matter is a "process" or "machine" but also whether it is "new" within the meaning of the statute.[1]

---

[1] 35 U.S.C. § 101 provides that:

> Whoever invents or discovers any *new . . . process, machine,* manufacture, or composition of matter . . . may obtain a patent therefor . . . .

"In construing a statute we are obliged to give effect, if possible, to every word Congress used. United States v. Menasche, 348 U.S. 528, 538-539 . . . ." *Reiter* v. *Sonotone Corp.*, 442 U.S. 330, 60 L.Ed.2d 931, 937 (1979).

The claim in each case is for a combination of elements in which the only novel feature is a computer program implementing an algorithm.[2]

Parker v. *Flook*, 437 U.S. 584 (1978), explains how such combination claims are to be analyzed for the presence of subject matter "eligible for patent protection" under § 101. 437 U.S. at 588. Pursuant to that analysis, "the algorithm . . . is treated as though it were a familiar part of the prior art" because it "reveals a relationship that has always existed" (437 U.S. at 591-2, 593, n. 15). "[O]nce that algorithm is assumed to be within the prior art, the application, considered as a whole" is then examined to determine whether it "contains [a] patentable invention" (437 U.S. at 594).

In each of these cases, the PTO Examiner and the Board of Appeals applied the *Flook* analysis to hold that the claims at issue do not define a patentable invention.

---

[2] In No. 79-855, the claim is for a "computer system" including a "data structure for storing coded signals" comprising four elements generically recited as "means" for storing a series of basic instructions as a microprogram. The Examiner found that the only novel aspect of the claimed invention resided in an algorithm designed to control the computer to solve a particular problem (Pet. App. D 33a-39a). The PTO Board of Appeals agreed that "save for the microprogramming", the computer system and data structure were old in the art (Pet. App. C 26a-27a). The CCPA left standing the Board's determination that the apparatus was old in the art and that only the microprogramming was new (Pet. App. A 13a).

In No. 79-1112, the claim is for a combination of steps including a mathematical algorithm, said to comprise a "method" for curing rubber. The CCPA did not reverse the PTO Board of Appeals' factual finding that everything in the claim except the use of the computer to calculate the cure time was old in the art (Pet. App. A 13a). See also Pet., p. 8, n. 5.

In each case, the CCPA tacitly acknowledged that if the *Flook* analysis is correct, the Board must be affirmed. Accordingly, the CCPA, in order to reverse the Board in each case, had to reject *Flook*, which it did, on the ground that "[w]hile the Supreme Court in that case may have found that analysis . . . convenient . . . we do not believe the Court meant to establish that analysis as a general test in determining compliance with § 101". *Application of Diehr*, 602 F.2d 982, 987, n. 6 (1979).

ARGUMENT

I. THE *FLOOK* ANALYSIS IS CORRECT

The *Flook* analysis for determining whether combination claims reciting an algorithm are patentable subject matter under § 101 is correct.

A. 35 U.S.C. § 101

Section 101 permits patents only for "new" subject matter which constitutes a "process, machine, manufacture, or composition of matter". The preclusion of patents for subject matter that is not "new" has "always existed in the statutory scheme" (*Graham* v. *John Deere Co.*, 383 U.S. 1, 12 (1966)).[3] It insures against breach of the "limits of the constitutional grant" which prohibit "Congress [from] authoriz[ing] the issuance of patents whose effects are to remove existent knowledge from the public domain" (*Ibid.*, at 6).

---

[3] The Court explained that:

> The precursors of these sections [35 U.S.C. §§ 101 and 102] are to be found in the Act of February 21, 1793, c. 11, 1 Stat. 318; Act of July 4, 1836, c. 357, 5 Stat. 117; Act of July 8, 1870, c. 230, 16 Stat. 198; Rev. Stat. § 4886 (1874).

Graham, supra, 383 U.S. at 14, n. 5.

The predecessor statute, R.S. 4886, set forth the provisions of § 101 and most of the provisions of § 102 in a single statute.[4]

As the Congressional committee reports state "Section 102 . . . includes, in effect, an amplification and definition of 'new' in section 101" which preserves the "language" of the old statute except for the substitution of "process" for "art".[5]

---

[4] R.S. 4886, subparagraphed for ease of consideration, states as follows:

> Any person who has invented or discovered any *new* and useful art, machine, manufacture, or composition of matter, or any *new* and useful improvements thereof, . . .
>
> not known or used by others in this country, before his invention or discovery thereof, and not patented or described in any printed publication in this or any foreign country, before his invention or discovery thereof, or more than one year prior to his application, and not in public use or on sale in this country for more than one year prior to his application, unless the same is proved to have been abandoned, may, upon payment of the fees required by law, and other due proceeding had, obtain a patent therefor. (Emphasis added.)

The Reviser's Note to § 101 states that:

> The corresponding section of existing statute [R.S. 4886] is split into two sections. . . .
>
> Section 101 follows the wording of the existing statute as to the subject matter for patents, except that . . . the word "art" has been replaced by "process", which is defined in section 100 . . . .

[5] S. Rep. No. 1979, 82d Cong., 2d Sess. 5, 6 (1952); House Report No. 1923, 82d Cong., 2d Sess. 6, 7 (1952).

Consistent with this legislative history, the CCPA has recognized that the word "new" in § 101 is to be given effect. In *Application of Bergstrom*, 427 F.2d 1394 (1970), the CCPA considered a rejection of the claims at issue "under § 101"

The imperative of § 101 that only "new" subject matter can be patented fully implements the prohibition against patents for "the basic tools of scientific and technical work" (*Gottschalk* v. *Benson*, 409 U.S. 63, 67 (1972)), without judicial compromise of the statute's plain and unambiguous language. This indeed is the essence of *Flook's* teaching.

Flook states that:

> The rule that the discovery of a law of nature cannot be patented rests, not on the notion that natural phenomena are not processes, *but rather on the more fundamental understanding that they are not the kind of "discoveries" that the statute was enacted to protect.* . . . (Footnote omitted; emphasis added.)

437 U.S. at 593.

The accompanying footnote 15 explains that this "fundamental understanding" is bottomed on the fact that natural phenomena, albeit processes, are *not "new"*, and for *that* reason "are not the kind of 'discoveries' " that are patentable. As the footnote states:

---

on the ground that the claimed compounds "are not 'new' within the connotation of the patent statute". 427 F.2d at 1397. The court's ruling was "that the subject matter claimed by appellants is 'new' ". 427 F.2d at 1402. In reaching that conclusion, the court held:

> [T]he criteria for determining whether given subject matter is "new" within the meaning of § 101 are no different than the criteria for determining whether that subject matter possesses the "novelty" expressed in the title of § 102. The word "new" in § 101 is defined and is to be construed in accordance with the provisions of § 102. Thus, that which possesses statutory novelty under the provisions of § 102 is also new within the intendment of § 101. . . . (Footnotes omitted.)

427 F.2d at 1401.

> The underlying notion is that a scientific principle . . . *reveals a relationship that has always existed.*
>
> "An example of such a discovery [of a scientific principle] was Newton's formulation of the law of universal gravitation. . . . But this relationship always existed—even before Newton announced his celebrated law. Such 'mere' recognition of a theretofore *existing* phenomenon or relationship carries with it no rights to exclude others from its enjoyment. . . . *Patentable subject matter must be new* (novel); not merely heretofore unknown. *There is a very compelling reason for this rule. The reason is founded upon the proposition* that in granting patent rights, the public must not be deprived *of any rights that it theretofore freely enjoyed.*" P. Rosenberg, Patent Law Fundamentals, § 4, at 13 (1975) (Emphasis added).

Flook, supra, 437 U.S. 593, n. 15.

The observations in *Flook* that subject matter which is a " 'process' in the ordinary sense of the word" may nevertheless be unpatentable because a "purely literal reading of § 101" (437 U.S. at 588-9) is foreclosed [6]

---

[6] The *Flook* opinion states:

> The plain language of § 101 does not answer the question. It is true, as respondent argues, that his method is a "process" in the ordinary sense of the word. But that was also true of the algorithm, which described a method for converting binary-coded decimal numerals into pure binary numerals, that was involved in Gottschalk v. Benson. The holding that the discovery of that method could not be patented as a "process" forecloses a purely literal reading of § 101. . . . (Footnote omitted.)

437 U.S. at 588-9.

The claim in *Benson* was unpatentable because it "would wholly pre-empt the [old] mathematical formula". 409 U.S. at 72.

accord with early decisions which recognize that, taken literally, the term "process" encompasses both patentable and unpatentable subject matter. As the opinion explains, however, the claim in *Flook* was unpatentable under § 101 for the fundamental reason that the subject matter was not "new".[7]

B. Apposite Decisions

At least two lines of precedent converge to require that each of the elements set forth in combination claims, however artfully drafted to refer to a "process", "machine" or "composition of matter", must be examined to detect any compromise of the public's unfettered right to everything which is old.

A first series of cases culminating in *Flook* and *Benson* holds that laws of nature, ideas, etc., are not patentable

---

[7] See, *e.g.*, *Corning* v. *Burden*, 56 U.S. (15 How.) 252 (1853), which points out that "process" has two distinct meanings, to wit:

(1) a "practicable method or means of producing a beneficial result or effect" (*id.* at 268) where "the result or effect is produced by chemical action, by the operation or application of some element or power of nature, or of one substance to another" (*id.* at 267); "[i]t is when the term process is used to represent the means or method of producing a result that it is patentable" (*id.* at 263);

(2) "in a more vague sense, in which it cannot be the subject of a patent . . . subjectively or passively as applied to the material operated on" (*id.* at 268)—*i.e.*, as a principle or abstract effect.

It was thus well settled under predecessors of § 101 that only the first category of "process" was patentable subject matter. Since the substitution of the word "process" for "art" in § 101 of the 1952 Act did not change the law, § 101 cannot be read literally to encompass both meanings of "process" above, but must be read to mean the first one, as *Corning* and its progeny hold.

and cannot impart patentability to an otherwise unpatentable combination of elements because they are not "new".

A second body of precedent holds that if an invention entails the improvement of only one element of an old combination, the improved element must be *separately* claimed and independently evaluated for compliance with the patent statutes, including § 101.

1. *Cases Concerning "Ideas"*

The two seminal principles of *Flook* and *Benson* and the line of decisions on which those opinions are bottomed are summarized in *Funk Bros. Seed Co.* v. *Kalo Inoculant Co.*, 333 U.S. 127, 130 (1948).

First is the truism that:

> He who discovers a hitherto unknown phenomenon of nature has no claim to a monopoly of it which the law recognizes. If there is to be an invention from such a discovery, it must come from the application of the law of nature to a new and useful end.

Second, a combination claim which purports to define "the application of the law . . . to a new and useful end" is not patentable if it is necessary to ["borrow] invention [*i.e.*, that which is 'new' and 'unobvious'] from the discovery of the natural principle itself." *Ibid.*, at 130, 132.[8]

---

[8] The entirety of the relevant passage from *Funk Bros.* states:

> But once nature's secret of the non-inhibitive quality of certain strains of the species of Rhizobium was discovered, the state of the art made the production of a mixed inoculant a simple step. Even though it may have been the product of skill, it certainly was not the product of invention. There is no way in which we could call it

Flook follows precisely the teaching of *Funk Bros.* and its antecedents. It refuses to impart patentability to an otherwise old process by "invention" "borrow[ed]" from a mathematical algorithm which was not "new". Instead, it correctly deemed the formula "to be within the prior art," and on that basis, concluded that "it is absolutely clear that respondent's application contains no claim of patentable invention" (437 U.S. at 594)—because all of the elements set forth in the claim, singly and in combination, were old.[9] See 437 U.S. at 594.

The *same* application of § 101 is explicated in *Anderson's-Black Rock, Inc.* v. *Pavement Salvage Co.*, 396 U.S. 57, 62 (1969). The claim at issue was for a paving machine made up of a combination of old elements in-

---

> such unless we borrowed invention from the discovery of the natural principle itself. . . .

333 U.S. at 132.

Congress was specifically importuned, without success, to adopt language in the 1952 Act which would overrule *Funk Bros.*

> so that it be made clear . . . and . . . specifically understood that where a discovery has been made which involves a natural principle and is useful and practically applicable, *but such application . . . does not in itself call for inventive ingenuity*, that it be then likewise recognized that an invention [which is patentable] has been made.

Testimony of I. J. Fellner, *Patent Law Codification and Revision, Hearings Before the Committee on the Judiciary, U.S. House of Representatives on H.R. 3760*, 82d Cong., 1st Sess. 116, 118 (June 13-15, 1951).

[9] An analysis which, as the Court's own n. 15 quoted *supra* makes clear, renders wholly immaterial the issue of whether the claimed method was a " 'process' in the ordinary sense of the word" (437 U.S. at 588).

cluding a radiant heat burner. The Court did not tarry over the fact that the claimed subject matter was manifestly a "machine". Instead, the Court went straight to the jugular to invalidate the patent. After quoting § *101* in its entirety, the Court held:

> *Absent here is the element "new"*. For as we have said, the combination patent added nothing to the inherent characteristics or function of the radiant-heat burner. (Emphasis added.)

396 U.S. at 62, n. 4.

2. *Combination Claims*

The issue in every case involving combination claims is whether the invention "produce[s] a 'new and different function,' Lincoln Co. v. Stewart-Warner Corp., 303 U.S. 545, 549, 82 L.Ed. 1008, 1010, 58 S. Ct. 662, within the test of validity of combination patents." *Anderson's-Black Rock, supra*, 396 U.S. at 60.

That "test of validity" makes it clear that a combination claim *must* be analyzed—as this Court analyzed the claim before it in *Flook*—to determine which elements, *if any*, are "new" and which are old. If the only novelty is in an improvement of one element in an old combination, that improved element must be *separately* claimed so that it may be precisely determined whether the only "invention" constitutes patentable *subject matter* (§ 101), and, *if so*, whether it is of sufficient importance (see 35 U.S.C. § 103) to warrant the grant of a lengthy private monopoly. As this Court put it in *Lincoln Engineering Co.* v. *Stewart-Warner Corp.*, 303 U.S. 545 (1938):

> As we said of Gullborg in the Rogers Case, having hit upon this improvement he did not patent it as such but attempted to claim it in combination with other old elements which performed no new function in his claimed combination. The patent is therefore void as claiming more than the applicant invented.

> . . . [T]he improvement of one part of an old combination gives no right to claim that improvement in combination with other old parts which perform no new function in the combination.
>
> * * * *
>
> We conclude that Butler's effort, *by the use of a combination claim, to extend the monopoly of his invention* of an improved form of chuck or coupler *to old parts or elements having no new function when operated in connection with the coupler renders the claim void.* (Emphasis added.)

303 U.S. at 549-50, 552.

Lincoln Engineering cites twenty-four cases (see 303 U.S. at 550, n. 6) which support this rule—included are three earlier decisions of this Court. The same rule has been applied by post-1952 cases [10] and is currently enforced by the PTO.[11]

---

[10] See, *e.g., Avery Products Corp.* v. *Morgan Adhesives Co.*, 496 F.2d 254 (6 Cir. 1974); *Pierce* v. *Ben-Ko-Matic*, 310 F.2d 475 (9 Cir. 1962); *Rohr Aircraft Corp.* v. *Rubber Teck, Inc.*, 266 F.2d 613 (9 Cir. 1959); *Bauer Patent Corp.* v. *Westinghouse Electric Corp.*, 193 F.Supp. 868 (D.N.C. 1961).

[11] The current Manual of Patent Examining Procedure states that:

> A rejection on the ground of old combination should be made whenever proper. . . . The fact that an applicant has improved one element of a combination which may be *per se* patentable does not entitle him to a claim to the improved element in combination with old elements where the elements perform no new function in the claimed combination. In re Hall, 41 CCPA 759.
>
> * * * *
>
> See also Lincoln Engineering Co. v. Stewart-Warner Corp., 303 U.S. 545, 37 USPQ 1 (1938); In re McCabe, 48 CCPA 881, 129 USPQ 149 (1961) (discussion of claim 13); and particularly In re Bernhart, 57 CCPA 737, 163 USPQ 611 (1969). (§ 706.03(j))

C. The *Flook* Analysis and § 103

In *Flook*, the Court was careful to explain that its analysis of what is patentable *subject matter* under § 101 is "not at all inconsistent with the view," borrowed from § 103, "that a patent claim must be considered as a whole".[12] 437 U.S. at 594.

To the contrary, analysis to identify the old and new among the elements recited in combination claims as variously applied in *Funk Bros.*, *Lincoln Engineering* and *Flook* finds statutory mandate in § 103 itself.[13]

Under that statute, the "subject matter as a whole" is to be evaluated in light of the "differences," if any, between the claims and the prior art.

The Congressional committee reports explain that:

> An invention which has been made, and which is new [as required by § 101] in the sense that the same

---

[12] Note 16, 437 U.S. at 594, of the *Flook* opinion states:

> Section 103, by its own terms, requires that a determination of obviousness be made by considering "the subject matter as a whole." 35 USC § 103 [35 USCS § 103]. Although this does not necessarily require that analysis of what is patentable *subject matter* under § 101 proceed on the same basis, we agree that it should. (Emphasis in original).

[13] 35 U.S.C. § 103 states:

> A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the *subject matter as a whole* would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made. (Emphasis added.)

> thing has not been made before, may still not be patentable if the *difference* between the *new* thing and what was known before *is not considered sufficiently great to warrant a patent.* . . . (Emphasis added.)[14]

In accord with the legislative history and the language of the statute itself, this Court repeatedly has held that

> Under § 103, the . . . *differences between the prior art and the claims at issue are to be ascertained*; and the level of ordinary skill in the pertinent art resolved. Against *this background*, the obviousness or nonobviousness of the subject matter is determined. (Emphasis added).[15]

To identify the "*differences*", as required by § 103, claims *must* be analyzed, as in *Flook*, to determine which, if any, of the elements are "new" and which are old. Only then is it possible to determine whether such "diffences" as may exist constitute subject matter, viewed in the context of the claim as a whole, which is patentable under § 101 and not "obvious" under § 103.

D. The *Flook* Analysis and the "Art" of Claim Drafting

The principle which is common to both of these lines of judicial precedent and also to § 103 is that the language of patent claims generally and claims to combinations in particular must be critically examined to insure that the public's interest is not compromised. The Court made the same point more recently in the context of § 101's "useful" prerequisite to patentability:

---

[14] S.Rep. No. 1979, 82d Cong., 2d Sess. 6 (1952); H.R.Rep. No. 1923, 82d Cong., 2d Sess. 7 (1952).

[15] *Graham* v. *John Deere Co.*, 383 U.S. 1, 17 (1966) cited and quoted in *Anderson's-Black Rock, Inc.* v. *Pavement Salvage Co.*, 396 U.S. 57, 62 (1969); *Dann* v. *Johnston*, 425 U.S. 219, 226 (1976); *Sakraida* v. *Ag Pro*, 425 U.S. 273, 280 (1976).

> [I]n light of the highly developed art of drafting patent claims so that they disclose as little useful information as possible—while broadening the scope of the claim as widely as possible—the argument based upon the virtue of disclosure must be warily evaluated. . . .

Brenner v. *Manson*, 383 U.S. 519, 534 (1966).

And, in *Flook*, this Court admonished that "[a] competent draftsman could attach some form of post-solution activity to almost any mathematical formula" (437 U.S. at 590), but to "make the determination of patentable subject matter depend simply on the draftsman's art . . . would ill serve the principles underlying the prohibition against patents for 'ideas' or phenomena of nature" (437 U.S. at 593).

In both *Flook* and *Lincoln Engineering*, analysis of the combination claims was necessary to identify which of the elements, if any, were new. By this means, the Court was able to determine whether, by artful drafting, unpatentable subject matter had been improperly monopolized. In *Lincoln Engineering*, the unpatentable subject matter was "the *old* parts of elements" of the claimed combination; in *Flook*, as in the two cases now before the Court, the unpatentable subject matter is an algorithm.

II. THE CLAIMS AT ISSUE

In these cases, as in *Flook*, analysis of the combination claims to identify the old and new elements, if any, makes it "absolutely clear that respondent's application[s] contain . . . no claim of patentable invention". 437 U.S. at 594.

In No. 79-1112, the claim is for an alleged improvement of only one of an old combination of process steps for the curing of rubber. The sole "difference" alleged to distinguish the respondent's invention, as a whole,

from the prior art is an unpatentable improvement in the method of calculating the cure time—a feature which "cannot support a patent" because there is no "other inventive concept in its application". *Flook, supra,* 437 U.S. at 594.

If the improved step in the old combination had been independently claimed, as controlling precedent from this Court requires, its unpatentability under § 101 because the subject matter is not "new" would have been facially apparent. See *Lincoln Engineering, supra; Anderson's-Black Rock, supra.*

In No. 79-855, the concurrent finding of the Patent Office tribunal which the CCPA left standing is that "save for the microprogramming" implemented by an algorithm, the claimed subject matter defined by the combination claims is old.[16] The claim, moreover, is in "Jepson" form, and the CCPA has agreed that the preamble of such a claim is to be disregarded in determining subject matter patentability under § 101.[17] All that is left of the claim is the recitation of four "means" which comprise the unpatentable microprogramming feature of the otherwise old apparatus.

---

[16] See n. 2, *supra,* p. 4.

[17] In its March 27, 1980 opinion in *Application of Walter,* —— F.2d ——, the CCPA states:

> Appellant's claims, except claims 10-12, are in Jepson [*Ex parte Jepson,* 1917 C.D. 62, 243 O.G. 525] format. The claim preambles merely set forth the environment in which the improvement operates. They show only the context in which the mathematical exercise in the claims are to be used. In each claim, what is positively claimed, as distinguished from environment, is "[a mathematical exercise] of correlating" (claim 7), or "the improved method of cross-correlating" (claim 16). The same is true of claims 11-12, drafted in illusory apparatus format.

Slip op., p. 24.

The CCPA, in its decisions in these cases, does not contend otherwise. Instead, these opinions and the subsequent opinion in *Walter* simply refuse to follow *Flook*.

III. THE CCPA DECISIONS

As the petitions for certiorari make clear, the CCPA in these and subsequent cases has refused to apply the analysis of claims required under *Flook* and its antecedents by limiting *Flook* to its facts.[18] Its unsupported rationale has been the naked emotionalism that if the *Flook* approach "were to be adopted it would immeasurably debilitate the patent system" (*Walter*, *supra*, slip. op., p. 15). In short, the CCPA has attempted to read the imperative of "new" out of § 101, and insisted that the "subject matter as a whole" language, borrowed from § 103 but not recited in § 101, renders it "impermissible" to separate a "claim into old and new elements" as a prerequisite to determining whether the subject matter is patentable within the meaning of § 101. *Application of Diehr*, *supra*, 602 F.2d 988.

A. Disregard of § 101's "New" Imperative

The established rules of statutory construction prohibit judicial elision of *any* word or expression appearing in a duly enacted law. See *Weinberger* v. *Hynson, Westcott & Dunning*, 412 U.S. 609, 633 (1973), wherein "Hynson's argument . . . [which] would render clause (C) superfluous" was repudiated by this Court because it "offends the well-settled rule of statutory construction that all parts of a statute, if at all possible, are to be given effect." To the same effect, see *Reiter* v. *Sonotone Corp.*, 442 U.S. 330, 60 L.Ed.2d 931, 937 (1979) in

---

[18] "[W]e do not believe the Court meant to establish that analysis as a general test in determining compliance with § 101. . . ." *Application of Diehr*, *supra*, 602 F.2d at 987, n. 6.

which the Court disapproved a "strained construction [which] would have us ignore the disjunctive 'or' and rob the term 'property' of its independent and ordinary significance; . . . [and] convert the noun 'business' into an adjective" because "[i]n construing a statute we are obliged to give effect, if possible, to every word Congress used."

Disdaining this fundamental rule and its own duty, in *In re Bergy*, 596 F.2d 952, 960, 962-964 (*cert. granted* No. 79-136, Oct. 29, 1979), the CCPA attacked *Flook* on the incredible premise that:

> *Prior art is irrelevant to the determination of statutory subject matter under § 101.* An invention can be statutory subject matter and be 100% old. . . . (Emphasis by the court).

This same unstable premise runs through the CCPA's ruling in No. 79-1112:

> Considerations of novelty . . . have no bearing on compliance with § 101. In re Bergy, 596 F.2d 952, 960-61, 962-63, 201 USPQ 352, 361, 362 (Cust. & Pat. App. 1979); Nickola v. Peterson, 580 F.2d 898, 906-907, 198 USPQ 385, 395-96 (6th Cir. 1978). . . .[19]

* * * *

---

[19] The *Nickola* opinion for the Sixth Circuit was written by the CCPA's Chief Judge Markey. Although the case was argued on November 30, 1977, it was not filed until June 23, 1978. Even *Nickola* does *not* hold, as *Bergy* implies, that "[c]onsiderations of novelty . . . have no bearing on compliance with § 101". Instead, *Nickola* acknowledges:

> That section [101] does employ the term "new," a synonym for "novel." . . . The words "new" and "useful," appearing twice, merely indicate the broad concept that a patent may not be obtained on that which is *old* [*e.g.*, an

> The novelty or otherwise of any element or even of all the elements or steps, or of the *combination has no bearing on whether the process is encompassed by § 101.*
>
> * * * *
>
> Novelty considerations have no bearing on whether claims define statutory subject matter under § 101. (Emphasis added.)

Application of Diehr, 602 F.2d at 987-89.

In its March 27, 1980 opinion in *Application of Walter*, *supra*, the CCPA, per Judge Rich, espouses the same heresy:

> Thus, if an inventor succeeds in applying scientific truth in a specific manner, resulting in the *improvement* of a process, machine, manufacture, or composition of matter, his invention is statutory, subject to the caveat that the underlying subject matter which has been improved *is itself within the bounds of § 101.* (Emphasis in the original.)

Slip. op., p. 16.

This approach is wrong. It ignores the fact that to be "within the bounds of § 101", an "improvement of a process, machine, manufacture, or composition of matter" must also be "new". If, as in *Flook*, the *only* "new" thing—the sole "difference" from the prior art; see § 103—is, in fact, a "scientific truth" which "has always

---

> algorithm or other scientific principle which has always existed] or useless. . . .

580 F.2d at 906.

It is one thing to point out, as in *Nickola*, that the requirement for novelty is more specifically defined in § 102—but quite another to hold, as the CCPA does, that the word "new" in § 101 is to be ignored in the application of that statute; particularly so, when the subject matter at issue is, by definition, "old" obviating any relevance of § 102's more particularized novelty conditions.

existed", *Flook*, *supra*, 437 U.S. at 593, n. 15, the improvement is not patentable subject matter under § 101 because it is *old*.

B. The *Flook* Analysis and the Subject Matter as a "Whole"

Flook's argument that "analysis of what is patentable subject matter under § 101" should proceed on the basis of considering the "subject matter as a whole", see n. 12, *supra*, p. 14, is now used by the CCPA to launch yet another ill-conceived attack on the *Flook* analysis. Notwithstanding this Court's careful explanation of why its consideration of each of the elements of the claims in *Flook* was "not at all inconsistent with the view that a patent claim must be considered as a whole" (437 U.S. at 594), the CCPA in No. 79-1112 *reversed* stating:

> [T]he fact that certain limitations in a claim may be novel and certain others may be old is irrelevant. . . . The focus of the inquiry should be [on] the claim, *as a whole*. . . . Flook, 437 U.S. at 594, n. 16. . . . (Emphasis in original.)

Application of Diehr, *supra*, 602 F.2d at 987. See also *Application of Walter*, *supra*, slip op., pp. 17, 19.

It is sufficient to note that this insistence upon a monolithic treatment of patent claims clashes with settled law, including the plain language of § 103 in which, alone, the "subject matter as a whole" passage appears.

C. "Debilitation" of the Patent System

Flook is not a revolutionary or unusual decision. It is simply the latest in a long line of consistent precedents by which this Court has always held that subject matter which effectively preempts a scientific principle or law of nature is not patentable. This standard, like that for "unobviousness", i.e., invention, "has remained invariable in this Court" (*Graham*, *supra*, 383 U.S. at

19) without adverse effect on the sound operation of the patent system.[20]

To give color to its groundless debilitation charge, the CCPA in *In re Sherwood*, 613 F.2d 809, 204 USPQ 537, 545, n. 11 (1980) and in *Walter*, *supra*, invites the false inference that the *Flook* analysis (labelled in *Walter* as "the 'point of novelty' approach", slip op., p. 17) would "condemn . . . as nonstatutory" a claim sustained as valid in *Mackay Radio & Telegraph Co.* v. *Radio Corp. of America*, 306 U.S. 86 (1939).

Per contra, *Mackay Radio* made *no* holding of claim validity, but instead put "the question of patentability to one side as a preface to [its] analysis of the infringement issue." *Flook*, *supra*, 437 U.S. at 591. As the *Mackay* Court put it:

> [W]e do not stop to solve the problem whether it was more than the skill of the art to combine the teaching of Abraham with that of Lindenblad and others who had pointed out that the arrangement of the wires at an angle enhanced directional radio activity along their bisector. We assume, without deciding the point, that this advance was invention even though it was achieved by the logical application of a known scientific law to a familiar type of antenna. . . .

306 U.S. at 94.

Far from making the affirmative holding of validity erroneously inferred by the CCPA, *Mackay* was at pains

---

[20] Even more to the point, the *Flook* analysis can be viewed as "immeasurably debilitat[ing]" or "potentially destructive" *only* insofar as its strict application would prevent the issuance of substantial numbers of patents. The question of what is "debilitating" or "destructive", so understood, is a *legislative* and not a judicial question, and should no more be addressed by the CCPA than by this Court.

to make known this Court's own grave doubts as to *that* issue and to emphasize that "*if* this *assumption* [of validity] is correct the invention was a narrow one . . . to be strictly construed . . . ." *Mackay, supra*, 306 U.S. at 94. In short, the opinion strongly suggests an analysis under § 101's antecedent statutes [21] that invalidates the *Mackay* claims.

Walter, supra, takes an equally misdirected tack in asserting that the *Flook* analysis would also condemn "the claim at issue in *Eibel Process Co.* [v. *Minnesota & Ontario Paper Co.*, 261 U.S. 45 (1923)]. The 'point of novelty' there was the application of the law of gravity to an otherwise well-known machine" (slip op., p. 16, n. 7). Careful reading of *Eibel*, however, shows that this Court did *not* view the "point of novelty" in the simplistic manner now espoused by the CCPA. Rather, the Court saw the *Eibel* invention as involving a coordination of several different factors to achieve a greatly improved machine. It noted, *inter alia*, that in attempting to improve the old machine, "there is no means, short of actual experiment, to enable one to anticipate results, and it is quite impossible to apportion to each factor its real influence" (261 U.S. at 62), and held that "what he [Eibel] saw and did was not obvious, and did involve discovery and invention" (261 U.S. at 68).

---

[21] Such an analysis was specifically applied later to invalidate the claim at issue in *Funk Bros.*

CONCLUSION

Discharge of the Court's "duty to construe the statutes as they now read in light of [its] prior precedents" and to "proceed cautiously when . . . asked to extend patent rights" (*Flook, supra*, 437 U.S. at 596) requires reversal of the decisions of the CCPA.

Respectfully submitted,

EDWARD S. IRONS
MARY HELEN SEARS
IRONS AND SEARS
1785 Massachusetts Avenue, N.W.
Washington, D.C. 20036
(202) 466-5200

ROBERT P. BESHAR
One New York Plaza
New York, New York 10004
(212) 558-0206

see 79
1112